

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



تسطیح اراضی

پروژه نرم افزار تسطیح اراضی کشاورزی

گردآوری: نبی کرملی زاده
پروژه پایان تحصیلی - گروه آبیاری
شماره دانشجویی: 8053081
استاد راهنما: آقای دکتر عصاره
دانشگاه آزاد اسلامی واحد اهواز
مهر ماه 1385
<http://www.Tastih.Nabi.ir/>

فهرست مطالب

صفحه	عنوان
1	1- پیشگفتار
3	2- مقدمه
5	3- تسطیح به روش حداقل مجذورات
6	4- معرفی و نحوه کار با نرم افزار
18	5- حل یک مثال به کمک نرم افزار
25	6- نکات برنامه نویسی نرم افزار
28	7- سورس کامل نرم افزار
61	8- منابع مورد استفاده

* یک عدد CD ضمیمه پروژه می باشد.

پیشگفتار

مطالبی که در این پروژه از پیش روی خواهید گذراند، در ارتباط با تسطیح اراضی کشاورزی می باشند.

در این مجموعه، مطالب اجمالی در ارتباط با تعریف و لزوم عمل تسطیح و شیوه تسطیح اراضی کشاورزی به روش حداقل مجذورات به صورت تئوری مورد بررسی قرار گرفته و سپس سورس¹ و شرح کامل نرم افزار² کامپیوتری که به زبان ویژوال بیسیک نسخه 6³ طراحی و نوشته ام را مشاهده خواهید کرد. به طور کلی توسط این نرم افزار کلیه عملیات مسطحاتی زمین به روش حداقل مجذورات و همچنین رسم نقشه توپوگرافی زمین قابل محاسبه و ترسیم می باشد.

امکانات و توانایی های نسخه نخست این نرم افزار بدین شرح می باشد:

- محاسبه و ترسیم نقشه توپوگرافی زمین به تفکیک رنگ.
- انجام محاسبات تسطیح زمین به روش حداقل مجذورات و امکان خروجی منظم به همراه امکان چاپ.
- رسم نقشه تسطیح با مشخص نمودن مناطق خاکریزی و خاکبرداری با رنگهای متمایز.
- محاسبه حجم خاکبرداری و خاکریزی.
- امکان گرفتن خروجی و ذخیره سازی به صورت فایل متنی یا تصویری.
- ذخیره سازی و بازیابی مجدد داده های کل پروژه.
- محیطی ساده، کاربرپسند و کاملاً فارسی.
- و ...

همان طور که ذکر شد، این نرم افزار کامپیوتری توسط زبان قدرتمند ویژوال بیسیک طراحی و برنامه نویسی شده است. در این نرم افزار سعی شده تا تمام اصول برنامه نویسی رعایت گردد و تمامی خطاهای کاربر⁴ پیشبینی شود تا در صورت لزوم پیغام مناسب جهت راهنمایی کاربر نمایش داده شود.

1 Source

2 Software

3 Visual Basic 6

4 Error

همچنین نرم افزار دارای محیطی¹ کاملاً کاربرپسند² و به صورت کاملاً فارسی می باشد که کار با آن را توسط عموم افراد به عنوان یک نرم افزار کاربردی ساده و آسان می نماید.

از آنجا که این نرم افزار جنبه آموزشی و تحقیقاتی داشته به صورت سورس باز³ ارائه می شود. اما هرگونه استفاده تجاری از آن ممنوع می باشد.

این نرم افزار دارای صفحه اختصاصی بر روی شبکه جهانی اینترنت به آدرس <http://www.Tastih.Nabi.ir/> می باشد. که جدیدترین نسخه های نرم افزار در این وب سایت قابل دریافت و بارگذاری⁴ می باشند.

در همینجا جا دارد از استاد راهنمای بنده جناب آقای دکتر علیرضا عصاره تشکر و قدر دانی نمایم.

همچنین از استاد ارجمند جناب آقای دکتر نوری که راهنمای بنده بودند کمال سپاس را دارم.

نبی کرمعلی زاده

www.Nabi.ir

-
- 1 Interface
 - 2 User Friendly
 - 3 Open Source
 - 4 Download

مقدمه

آماده نمودن زمینهای ناهموار برای آبیاری به روش سطحی مستلزم ایجاد تغییراتی در وضع پستی و بلندی طبیعی زمین است. در نتیجه این تغییرات که اصطلاحاً به آن "تسطیح"¹ گفته میشود، سطح طبیعی زمین تبدیل به سطحی مناسب نوع آبیاری و نیز جنس خاک میگردد. برای یک قطعه زراعی این سطح ممکن است از یک صفحه مستوی با شیبهای یکنواخت در جهت های آبیاری و عمود بر آن و یا از چندین صفحه مستوی با شیب های متفاوت تشکیل شده باشد.

تغییر در پستی و بلندی های زمین در بسیاری از پروژه های عمرانی لازم و ضروری است. از جمله این پروژه ها می توان شهرسازی، ایجاد استخر و دریاچه مصنوعی به منظور های مختلف (ذخیره آب، پرورش ماهی، ورزشهای آبی و غیره)، برداشت از معادن سطحی و بالاخره تسطیح زمین به منظور کشاورزی را نام برد.

در کلیه موارد بالا پس از برداشت پستی و بلندی های طبیعی زمین و تهیه نقشه منحنی های تراز² آن، طرح پروژه مورد نظر ریخته می شود. سپس ارتفاع نقاط بعد از تغییر توپوگرافی و از روی آن عمق خاکبرداری و ارتفاع خاکریزی در نقاط مختلف شبکه و هم چنین حجم عملیات خاکی³ محاسبه می گردد. در مرحله بعد عمق های خاکبرداری و ارتفاعات خاکریزی، برای رانندگان ماشین های خاکبرداری و تسطیح در روی زمین مشخص می شود. آخرین مرحله اجرای پروژه است. در این مرحله باید نظارت کامل به عمل آید تا پروژه مطابق محاسبات انجام شده اجرا گردد.

گرچه موضوع و مقصود اصلی این پروژه و تحقیق تسطیح اراضی به منظور کشاورزی و به خصوص آبیاری سطحی⁴ است، اما از مطالب آن میتوان در تغییر توپوگرافی زمین به منظور های دیگر مثل کارهای عمرانی نیز استفاده نمود.

آبیاری سطحی یکی از روشهای مهم و متداول آبیاری است. در این روش که انواع مختلفی دارد، سعی بر آن است که آب بطور یکنواخت در سطح زمین گسترش یابد. بنابراین طی عملیات تسطیح باید سطح زمین به یک سطح مستوی شیبدار تبدیل گردد. در اغلب روشها این سطح باید دارای دو شیب یکنواخت، یکی در جهت آبیاری و دیگری در جهت عمود بر آن باشد. مقدار این شیبها نیز باید متناسب با نوع آبیاری

1 Land Forming = Land Shaping = Land Grading = Land Leveling

2 Contour Line

3 Volume of Earthwork

4 Surface Irrigation

انتخاب گردد. آماده کردن زمین برای آبیاری سطحی علاوه بر تسطیح زمین با شیب معین احداث کانالهای آبیاری، در نظر گرفتن زهکش برای خارج کردن آب مازاد، مرزبندی بوسیله خاکریز های کم ارتفاع، تراس بندی و گاهی احداث جاده جهت دستیابی به مزارع را نیز شامل می شود.

تسطیح اراضی مستلزم مخارج زیادی است.¹ لذا جنبه اقتصادی آن را باید در نظر داشت. با توجه به حجم عملیات خاکی در هکتار و نیز فواصل خاک کشی میتوان مخارج تسطیح را، پیش از اجرای پروژه، تخمین زده آن پروژه را از نظر اقتصادی ارزیابی نمود.

اصطلاحات متفاوتی برای عملیات مختلف تسطیح وجود دارد. از آنجا که از این اصطلاحات ممکن است برداشت های متفاوتی بشود، در اینجا به تعریف آنها می پردازیم.

- 1- تسطیح: اصطلاحی است کلی که به تمام عملیات از جمله هموار کردن، شیب دادن، تراز کردن و صاف کردن زمین اطلاق می شود.
 - 2- هموار کردن: این اصطلاح به تسطیح تقریبی زمین اطلاق می شود. گاهی پستی و بلندی های زمین بقدری آشکار و مشخص است که بدون نقشه برداری می توان آنها را تشخیص داد و به هموار کردن زمین پرداخت.
 - 3- تراز کردن: به عملیاتی اطلاق می شود که طی آن سطح طبیعی زمین به یک سطح مستوی، که همه نقاطش هم ارتفاع هستند، تغییر می یابد.
 - 4- شیب دادن: به عملیاتی اطلاق می شود که طی آن سطح طبیعی زمین به سطح شیب دار تغییر می یابد. معمولاً شیب در دو جهت عمود بر هم متفاوت ولی در هر یک از این دو جهت یکنواخت است.
 - 5- صاف کردن: به عملیاتی اطلاق می شود که طی آن ناهمواری های مختصری که در سطح زمین باقی مانده است از بین رفته و سطح زمین به یک سطح کاملاً مستوی تغییر می یابد.
- صاف کردن در واقع مرحله نهایی عملیات تسطیح است.

1 در فاصله بین سالهای 65-1945 تنها در ایالت کالیفرنیا آمریکا 0,25 میلیارد دلار برای تسطیح زمین خرج شده است.

تسطیح به روش حداقل مجذورات

اولین بار در سال 1940، سی - وی - جیوان¹ از روش آماری حداقل مجذورات برای محاسبه بهترین شیب در تسطیح اراضی استفاده نمود. نامبرده با وجودی که فرمول های مربوط را در یک حالت کلی ارائه داد ولی از آنها فقط برای محاسبه بهترین شیب در زمینهای مستطیل شکل استفاده کرد. گرانیث - ای - چاک² در سال 1947 استفاده از این روش را برای زمینهای نامنظم تعمیم داد.

منظور از بهترین شیب در تسطیح یک قطعه زمین شیبی است که برای بوجود آوردن آن حداقل عملیات خاکی انجام گیرد و در عین حال بین حجم خاکبرداری و خاکریزی تعادل وجود داشته باشد.

روش حداقل مجذورات، ما را در پیدا کردن شیب به ترتیبی که حداقل عملیات خاکی انجام گیرد، یاری میدهد. برای حفظ تعادل بین حجم های خاکبرداری و خاکریزی، پس از یافتن بهترین شیب، بایستی سطح زمین پس از تسطیح را طوری انتخاب نمائیم که از مرکز ثقل سطح طبیعی زمین عبور نماید.

مطابق روش حداقل مجذورات، چنانچه از میان تعدادی نقطه در فضا، یک صفحه مستوی به ترتیبی عبور دهیم که مجموع مجذورات فواصل عمودی نقاط تا صفحه مینیمم باشد، صفحه مزبور بهترین صفحه برای نمایش رابطه خطی بین ارتفاع و طول و عرض این نقاط است. به عبارت دیگر این صفحه، من حیث المجموع، بیش از هر صفحه دیگری به این نقاط نزدیک است.

در تسطیح اراضی نقاط یاد شده در بالا در واقع رئوس شبکه می باشد. طول، عرض و ارتفاع این نقاط برای ما معلوم است. صفحه مستوی فوق الذکر نیز نشان دهنده سطح زمین پس از تسطیح است.

این بخش تئوری تسطیح به روش حداقل مجذورات بود. در اینجا از ارائه فرمول ها صرفنظر می شود اما کلیه آنها در ساخت و نوشتن نرم افزار، بکاربرده شده و لحاظ شده اند.

1 C.V. Givan
2 G.E. Chugg

معرفی و نحوه کار با نرم افزار

در اینجا به آموزش کار با نرم افزار می پردازیم.

پس از اجرای نرم افزار، با صفحه مشابه صفحه زیر مواجه می شوید.

جهت ورود به نرم افزار بر روی گزینه "اجرای برنامه" کلیک نمایید.



صفحه اصلی نرم افزار مانند تصویر زیر ظاهر خواهد شد.



اکنون به شرح بخشهای مختلف نرم افزار که بر روی شکل زیر نشان داده شده اند خواهیم پرداخت.



1 بخش شماره یک:

این بخش، زمین فرضی به صورت مشبک می باشد که می بایستی زمین را بر روی این شبکه مشبک شبیه سازی کنیم. محل برخورد هر یک از خطوط شبکه با یکدیگر یک گره محسوب شده که با کلیک بر روی آنها میتوان ارتفاع آن گره را برای نرم افزار تعریف کرد.

زمانی که نشانه موس را بر روی این بخش حرکت میدهید، نشانه موس به شکل دست (☞) تبدیل خواهد شد. که میتوانید با پایین نگه داشتن دکمه سمت چپ موس و حرکت همزمان موس به اطراف، زمینه را جابجا کرده و تمام فضای کار را براحتی مشاهده نمائید. این عمل شما را در استفاده از اسکرول های افقی و عمودی¹ بی نیاز خواهد ساخت. اصطلاحاً به این عمل، کشیدن و رها کردن یا درگ² کردن گفته می شود.

1 Horizon & Vertical Scroll
2 Drag & Drop

2 بخش شماره دو:

در این بخش مختصات گره فعال که اشاره گر موس¹ بر روی آن قرار گرفته است ظاهر میگردد.

3 بخش شماره سه:

این بخش نوار فهرست یا نوارمنو² نرم افزار جهت دسترسی به کلیه امکانات نرم افزار می باشد؛ که شامل منوی اصلی، تنظیمات، عملیات، ابزارها و راهنما است. بخشهای مختلف نوارمنو بدین شرح می باشد:

3-1 منوی اصلی:



3-1-1 پرونده جدید:

با این گزینه تمامی اطلاعات ورودی صفر شده و تنظیمات به حالت پیشفرض بازخواهد گشت، و نرم افزار برای پروژه جدید حاضر خواهد شد.

3-1-2 خواندن پرونده:

با انتخاب این گزینه و انتخاب فایل پروژه که پیش از این توسط همین نسخه از نرم افزار ذخیره گردیده است میتوانید کلیه داده ها و تنظیمات را درون نرم افزار وارد نمائید تا در صورت لزوم آنها را ویرایش کرده و محاسبات و نقشه های جدید را استخراج نمائید.

3-1-3 ذخیره سازی:

با استفاده از این گزینه می‌توانید کلیه داده‌هایی که در نرم افزار وارد کرده‌اید و تنظیماتی که اعمال کرده‌اید را در قالب یک فایل ذخیره‌سازی نمائید و در آینده آن را مجدداً به نرم افزار وارد نمائید.

3-1-4 خروج:

این گزینه جهت خروج از نرم افزار می‌باشد.

3-2 تنظیمات:



3-2-1 شبکه بندی:

در بخش طول و عرض، ابعاد زمین بر حسب واحد دلخواه را وارد می‌نمائید. توجه داشته باشید که کلیه واحدهای ورودی یکسان باشند. همچنین واحدهای خروجی معادل واحدهای ورودی خواهند بود. در صورتی که واحد مورد نظر شما بسیار زیاد و

یا بسیار کوچک باشد به طوری که در ترسیم و شبیه سازی آن در نرم افزار مشکلی ایجاد شود، با مقیاس مناسب و دلخواه ابعاد زمین را کوچکتر و یا بزرگتر نمایید. فواصل خطوط شبکه از یکدیگر در دو جهت محور X و Y را نیز در بخش مربوطه وارد نمایید.

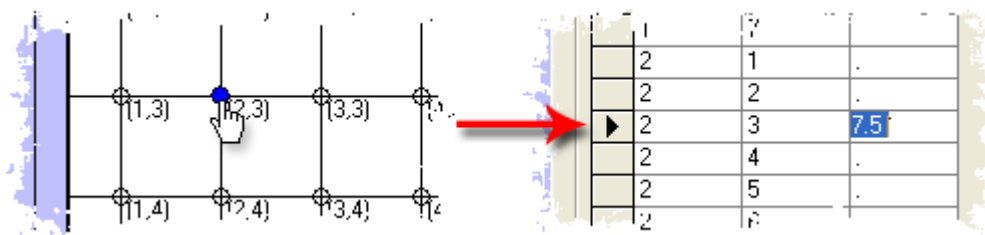
در صورتی که نقاط برداشت شده، از حاشیه زمین فاصله دارند، این فاصله را در دو جهت X و Y میتوانید وارد نمایید.

2-2-3 ارتفاعات:

X	Y	H
1	1	.
1	2	.
1	3	.
1	4	.
1	5	.
1	6	.
1	7	.
2	1	.
2	2	.
2	3	.
2	4	.
2	5	.
2	6	.
2	7	.
3	1	.
3	2	.

کلیه ارتفاعات نقاط برداشت شده را در این بخش می بایست وارد نمایید. دو ستون X و Y نشانه مختصات نقاط می باشند که قابل تغییر نمی باشند. شما تنها کافیست ارتفاع نقطه برداشت شده را در ستون H وارد نمایید.

جهت سرعت و سهولت در وارد نمودن مقادیر، میتوانید بر روی گره های واقع بر روی زمین فرضی کلیک نموده تا مکان نما¹ به مختصات مورد نظر در جدول انتقال یابد. در این لحظه تنها کافیست ارتفاع مورد نظر را تایپ نمایید. مانند تصویر زیر:



3-2-3 شیب/مرکز ثقل:

رنگها | شیب/مرکز ثقل | ارتفاعات | شبکه بندی

تنظیمات شیب/مرکز ثقل

☐ تصحیح شیب ها لحاظ شود؟

Sx: (,) %

Sy: (,) %

R =

☒ مختصات مرکز ثقل خودکار حساب شود؟

Xc =

Yc =

در صورتی که مایلید تصحیح مربوط به شیبها لحاظ شود، تیک مربوطه را فعال کرده و سپس حدود Sx و Sy را وارد نمایید.

در صورتی که تمایل دارید مختصات مرکز ثقل زمین به صورت خودکار حساب شود، تیک مربوطه را فعال کنید و در غیر این صورت آن را غیر فعال کرده و مقادیر مرکز ثقل را به صورت دستی وارد نمایید. این انتخاب بدین جهت در نظر گرفته شده است چون در محاسبه مختصات مرکز ثقل از روشهای گوناگون، خطاهای چشمگیری حاصل میشود. بنابراین این گزینه جهت آزمون به ازای مختصات مرکز ثقلهای متفاوت قرار داده شد.

همچنین مقدار R که همان نسبت مجموع میزان خاکبرداری به خاکریزی میباشد را در محل مشخصه وارد نمایید. مقدار R جهت آزمون یکسان بودن میزان خاکبرداری و خاکریزی کاربرد دارد و معمولاً به سمت 1 میل میکند. البته از آنجا که در حین عملیات خاکی و بر اثر رفت و آمد ماشین ها و کوبیده شدن خاک، زمین کمی نشست

مینماید. به صورت تجربی معمولاً اندکی بیشتر از 1 در نظر گرفته می شود تا میزان خاکبرداری از خاکریزی اندکی بیشتر شود و در عمل تسطیح با کمبود خاک مواجه نشویم.

3-2-4 رنگها:

#	فعال	ارتفاع	رنگ
1)	<input checked="" type="checkbox"/>	7.0	#FFC0C0
2)	<input checked="" type="checkbox"/>	7.2	#FFE0C0
3)	<input checked="" type="checkbox"/>	7.4	#FFFFC0
4)	<input checked="" type="checkbox"/>	7.6	#C0FFC0
5)	<input checked="" type="checkbox"/>	7.8	#C0FFFF
6)	<input checked="" type="checkbox"/>	8.0	#C0C0FF
7)	<input checked="" type="checkbox"/>	8.2	#FFC0FF
8)	<input checked="" type="checkbox"/>	8.4	#C0C0C0
9)	<input checked="" type="checkbox"/>	8.6	#FF8080

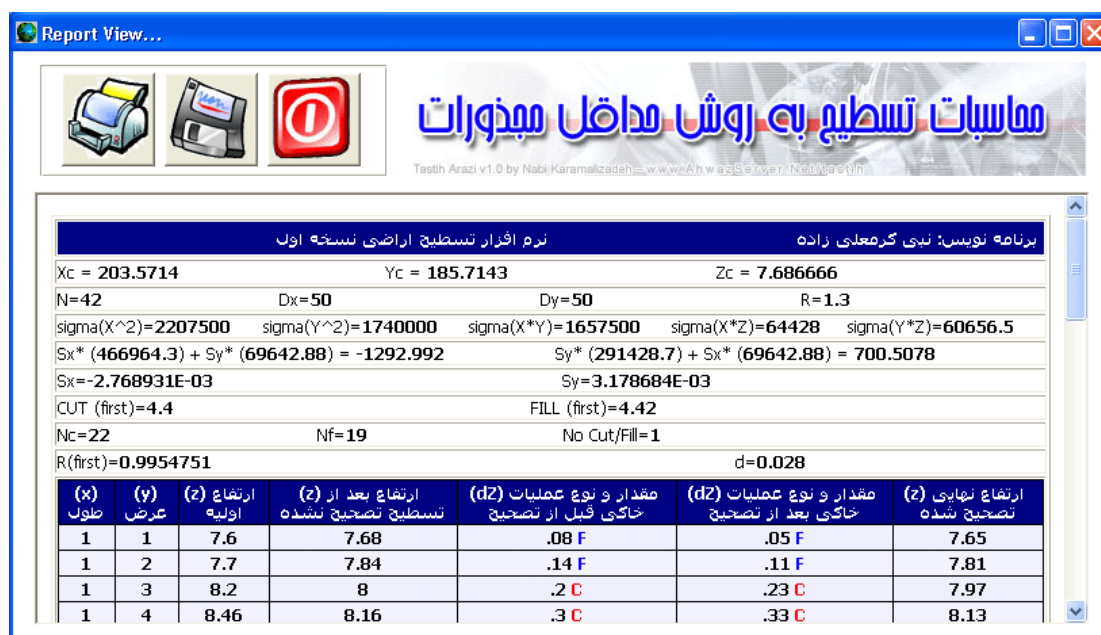
در این بخش میتوانید تا حداکثر 9 رنگ را تعریف نموده، این رنگها در رسم نقشه توپوگرافی زمین کاربرد دارند. برای این منظور کفایست در ستون ارتفاع، میزان ارتفاع دلخواه را وارد کرده و با کلیک بر روی دکمه مقابل آن، رنگ متناظر با آن ارتفاع را مشخص نمایید.

3-3 عملیات:

کلید	عملیات
Ctrl+T	انجام محاسبات تسطیح
Ctrl+M	رسم نقشه تسطیح
Ctrl+P	رسم نقشه توپوگرافی
Ctrl+R	نمایش نقطه مرکز ثقل
Ctrl+F	حجم کل خاکبرداری/خاکریزی

3-3-1 انجام محاسبات تسطیح:

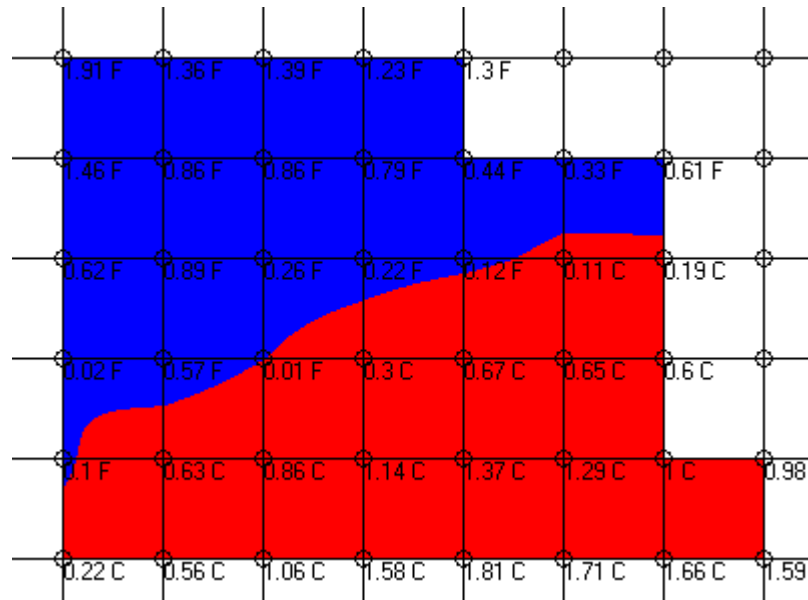
کلیه محاسبات و جداول مربوط به تسطیح زمین توسط این گزینه قابل محاسبه و نمایش می باشد. با انتخاب این گزینه، کلیه پارامترهای اولیه محاسبه و نمایش داده شده و سپس در جدولی به ترتیب، طول، عرض، ارتفاع اولیه، ارتفاع بعد از تسطیح تصحیح نشده، مقدار و نوع عملیات خاکی قبل از تصحیح، مقدار و نوع عملیات خاکی بعد از تصحیح و ارتفاع نهایی تصحیح شده نمایش داده میشود. در انتها نیز پارامترهای تصحیح شده و نهایی نمایش داده خواهند شد. که نمونه ای از آن در تصویر زیر آمده است:



در بالای پنجره محاسبات، سه دکمه به ترتیب جهت چاپ محاسبات و جداول، ذخیره در قالب یک فایل و دکمه بستن پنجره وجود دارد.

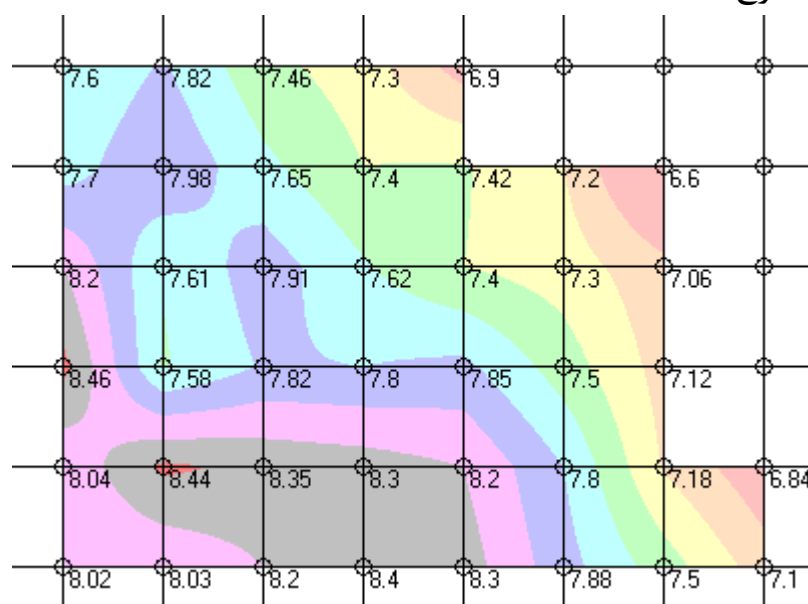
3-3-2 رسم نقشه تسطیح:

با انتخاب این گزینه، نقشه توپوگرافی زمین به تفکیک رنگ نمایش داده خواهد شد. که رنگ آبی نماینده مناطقی است که باید در آنها خاکریزی صورت گیرد و رنگ قرمز نماینده مناطقی از زمین می باشد که باید از آنها خاکبرداری صورت گیرد.



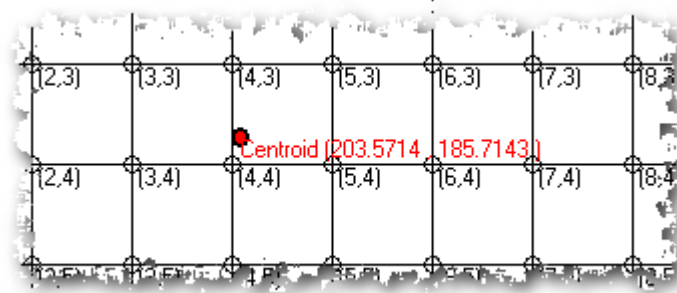
3-3-3 رسم نقشه توپوگرافی:

با انتخاب این گزینه نقشه توپوگرافی زمین ترسیم می گردد. که هر رنگ نماینده ارتفاعی خاص می باشد. نحوه انتخاب و تنظیم این ارتفاعات و نیز رنگهای نظیر آنها در بخش 3-2-4 شرح داده شد.



3-3-4 نمایش نقطه مرکز ثقل:

این گزینه موقعیت مرکز ثقل زمین به همراه مختصات آن نسبت به محورهای فرضی حاشیه زمین را نشان میدهد.



3-3-5 حجم کل خاکبرداری/خاکریزی:

این گزینه حجم خاکبرداری و خاکریزی کل را به روش میان یابی انجام خواهد داد. از آنجا که همیشه روش میان یابی همراه با مقداری خطا می باشد، در استفاده از این گزینه دقت فرمائید.

حجم کل خاکبرداری/خاکریزی

حجم کل خاکبرداری و خاکریزی منطقه بدین شرح می باشد،
نقاط بین گره ها به روش میان یابی محاسبه شده اند،
واحد ها برابر مکعب واحد مقادیر ورودی می باشند.

حجم خاکبرداری کل: 8347.56

حجم خاکریزی کل: 2840.36

بستن

3-4 ابزارها:



3-4-1 افزودن تصویر:

گزینه افزودن تصویر، به منظور قرار دادن یک تصویر در پس زمینه و به جای زمین فرضی می باشد. این کار به شبکه بندی زمین کمک کرده و میتوان نقاط و خطوط

فرضی را بر نقاط واقعی برداشت شده از روی زمین که در زمینه نرم افزار قرار داده اید منطبق کرد.

2-4-3 حذف تصویر:

اگر مایل به حذف تصویری که در پس زمینه نرم افزار قرار داده اید می باشید میتوانید از این گزینه استفاده نمائید. همچنین این گزینه جهت حذف نقشه های توپوگرافی و نقشه تسطیح ترسیم شده نیز کاربرد دارد.

3-4-3 ذخیره نقشه:

توسط این گزینه میتوانید نقشه توپوگرافی یا نقشه تسطیح زمین، تهیه شده توسط نرم افزار را به همراه خطوط و گره های ترسیمی به صورت یک فایل تصویری با فرمت BMP بر روی کامپیوتر خود ذخیره نمائید.

3-5 راهنما:



1-5-3 راهنمای برنامه:

این گزینه موجب نمایش راهنمای کامل نرم افزار به زبان فارسی خواهد شد.

2-5-3 درباره برنامه:

معرفی و توضیحات کوتاه درمورد این نرم افزار را میتوانید در این بخش مطالعه فرمائید.

4 بخش شماره چهار:

جهت ترسیم سریعتر نقشه توپوگرافی میتوانید از این گزینه استفاده نمائید. عملکرد این گزینه درست همانند گزینه مشروح در بخش 3-3-3 می باشد.

5 بخش شماره پنج:

جهت ترسیم سریعتر نقشه تسطیح میتوانید از این گزینه استفاده نمائید. عملکرد این گزینه درست همانند گزینه مشروح در بخش 2-3-3 می باشد.

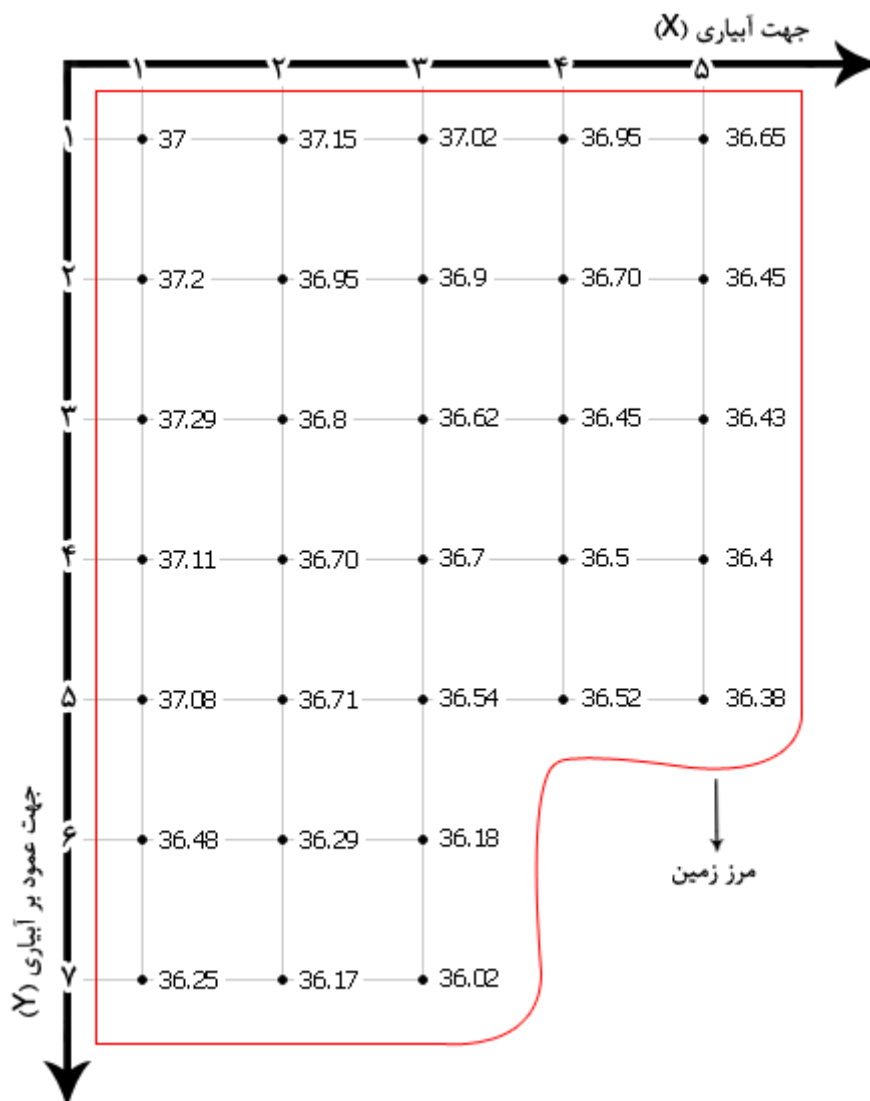
6 بخش شماره شش:

کلیه محاسبات تسطیح زمین ، جداول و فرمولهای بکار رفته با انتخاب این گزینه محاسبه و نمایش داده خواهد شد. عملکرد این گزینه درست همانند گزینه مشروح در بخش 1-3-3 می باشد.

حل یک مثال به کمک نرم افزار

مثال: خط قرمز مرز زمین فرضی را نشان می دهد. با فرضیات زیر زمین را تسطیح نموده، مقدار و حجم خاکبرداری و خاکریزی را تعیین نموده و نقشه توپوگرافی و نقشه تسطیح شده را رسم نمائید.

- فاصله نقاط شبکه $D=70$
- تعداد نقاط شبکه $N=31$
- $R=1.26$
- شیب در جهت آبیاری (0.1 – 0.3) درصد
- شیب در جهت عمود بر آبیاری (0.2 – 0.5) درصد



پاسخ: در صورتی که نقشه توپوگرافی واقعی را به ما داده باشند میتوان از منوی "ابزارها"، گزینه "افزودن تصویر" را انتخاب نمود و تصویر واقعی را در پس زمینه کار قرار داد.

قدم اول در حل مسئله تعیین ابعاد زمین فرضی می باشد. به توجه به تعداد نقاط و فواصل آنها، طول و عرض زمین 350 در 490 در نظر گرفته می شود. فاصله حاشیه زمین معمولاً نصف فاصله نقاط شبکه می باشد. در این مثال 35 در نظر گرفته میشود. با این توضیحات، بخش نخست در نرم افزار را با مقادیری که در تصویر مشاهده میکنید پر می نمائیم.

اکنون زمین فرضی ترسیم خواهد شد و با کلیک بر روی هر یک از نقاط شبکه ارتفاع نظیر آن را طبق صورت مسئله در نرم افزار وارد می نمائیم. مانند تصویر مقابل.

نکته: در مورد نقاطی که خارج از مرز زمین قرار دارند و یا ارتفاع آنها نامشخص می باشد مقدار صفر را وارد میکنیم. در این مثال برای نقاط (4و6)، (4و7)، (5و6)، (5و7) مقدار صفر وارد خواهیم کرد.

	X	Y	H
▶	3	6	36.18
	3	7	36.02
	4	1	36.95
	4	2	36.7
	4	3	36.45
	4	4	36.5
	4	5	36.52
	4	6	.
	4	7	.
	5	1	36.65
	5	2	36.45
	5	3	36.43
	5	4	36.4
	5	5	36.38
	5	6	.
	5	7	.

تنظیمات شیب/مرکز ثقل

☒ تصحیح شیب ها لحاظ شود؟

Sx: (0.1 , 0.3) %

Sy: (0.2 , 0.5) %

R = 1.26

☒ مختصات مرکز ثقل خودکار حساب شود؟

Xc = 0

Yc = 0

در بخش سوم همانطور که در تصویر مشاهده می کنید. گزینه مربوط به شیب را فعال کرده و مقادیر شیبها را بر حسب درصد وارد می نمائیم. همچنین مقدار R را در این بخش وارد می نمائیم.

در مورد مختصات مرکز ثقل زمین، توضیح اینکه این مختصات توسط نرم افزار بدین صورت محاسبه می شود که در هر ردیف، شماره ردیف میخ در تعداد نقاط همان ردیف ضرب شده و این عمل در مورد تمام ردیف ها انجام میشود. سپس مجموع اعداد بدست آمده به عنوان طول مرکز ثقل در نظر گرفته

خواهد شد. در مورد عرض مرکز ثقل به روش مشابه عمل خواهد شد.

توجه داشته باشید که روشهای دستی دیگری نیز جهت محاسبه مختصات مرکز ثقل وجود دارند از جمله تقسیم زمین به اشکال منظم هندسی و محاسبه میانگین وزنی مختصات مرکز ثقل تک تک اشکال. اما این روش با روش قبل تفاوتهایی از نظر مقدار دارد. در صورتی که مایلید به روش دستی مختصات مرکز ثقل را محاسبه نموده و سایر

محاسبات را به نرم افزار محول کنید گزینه "مختصات مرکز ثقل خودکار حساب شود؟" را غیر فعال نموده و مقادیر Xc و Yc را وارد نمائید. در غیر این صورت آن را به حالت فعال رها کرده تا نرم افزار آن را به صورت خودکار محاسبه نماید.

در بخش آخر تنظیمات، دامنه رنگ جهت رسم نقشه توپوگرافی را تنظیم میکنیم.

برای اولین رنگ، حداقل ارتفاع موجود در شبکه یعنی عدد 36,01 و برای آخرین رنگ، حداکثر ارتفاع موجود در شبکه یعنی عدد 37,29 را در نظر میگیریم و در مورد مابقی

تنظیمات رنگها

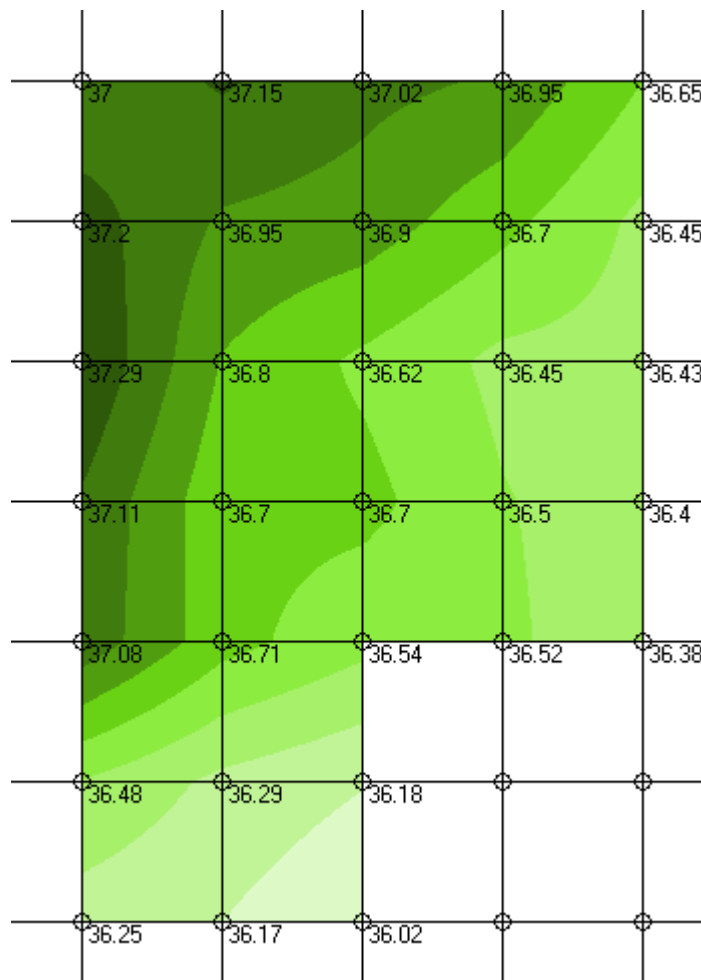
رنگ	ارتفاع	فعال	#
#F4FDEE	36.01	<input checked="" type="checkbox"/>	1)
#DDF9C6	36.17	<input checked="" type="checkbox"/>	2)
#C1F497	36.33	<input checked="" type="checkbox"/>	3)
#A7F06A	36.49	<input checked="" type="checkbox"/>	4)
#8DEC40	36.65	<input checked="" type="checkbox"/>	5)
#6AD215	36.81	<input checked="" type="checkbox"/>	6)
#509E10	36.97	<input checked="" type="checkbox"/>	7)
#3F7B0D	37.13	<input checked="" type="checkbox"/>	8)
#2D5909	37.29	<input checked="" type="checkbox"/>	9)

نقاط مابین از ارتفاعات دلخواه با رعایت گام مناسب استفاده می نمائیم. در این مثال، گام را 0,15 در نظر گرفته و ارتفاعات را به ترتیب 36,01 و 36,17 و 36,33 و... 37,29 قرار می دهیم.

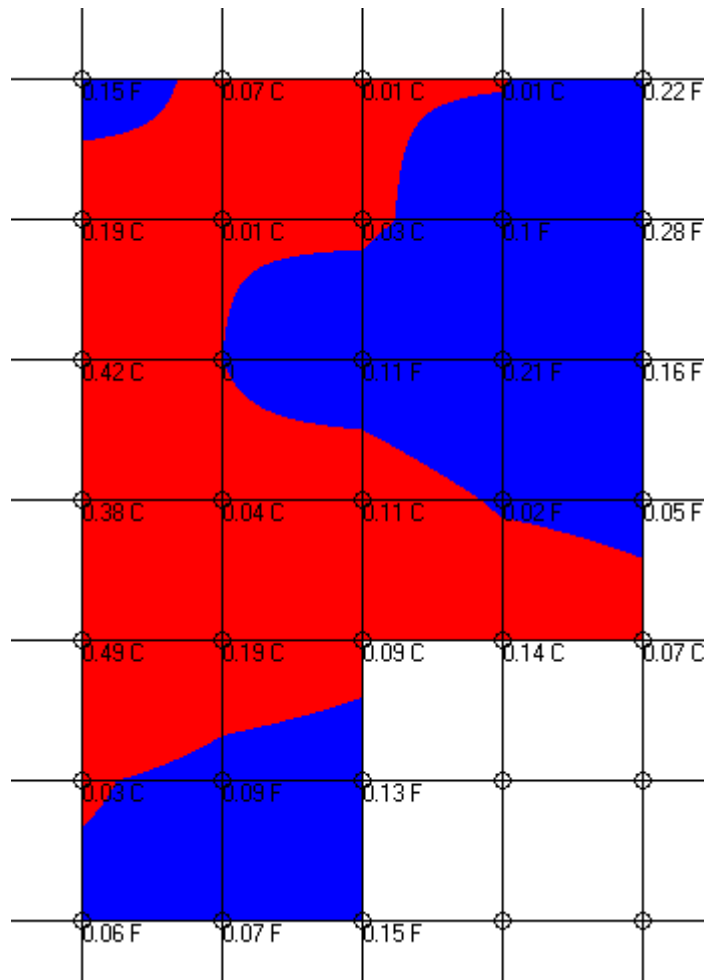
حداکثر از 9 رنگ متفاوت میتوان استفاده نمود. میتوانید برای نقاط با ارتفاع کمتر از رنگهای روشنتر و برای نقاط با ارتفاع بیشتر از رنگهای تیره تر استفاده نمائید. مانند تصویر.

ورود داده های خام به نرم افزار انجام پذیرفت. اکنون میتوانیم نتایج و محاسبات را مشاهده نمائیم.

جهت مشاهده نقشه توپوگرافی زمین بر روی گزینه "نقشه توپوگرافی" در صفحه اصلی نرم افزار کلیک کنید. تصویری مانند آنچه در زیر آمده است ترسیم خواهد گشت.



جهت مشاهده نقشه نشانگر نقاط خاکبرداری یا خاکریزی بر روی گزینه نقشه تسطیح در صفحه اصلی نرم افزار کلیک نمائید. نقشه ای مشابه نقشه زیر ترسیم خواهد شد که رنگ آبی نمایانگر نقاط و مناطقی است که باید در آنها خاکریزی صورت گیرد (FILL) و رنگ قرمز نمایانگر نقاط و مناطقی است که باید در آنها خاکبرداری صورت گیرد (CUT). میزان خاکبرداری یا خاکریزی در کنار هر نقطه نوشته شده است.



اکنون جهت انجام محاسبات تسطیح و مشاهده میزان خاکبرداری یا خاکریزی هر نقطه بر روی گزینه "محاسبات تسطیح" در صفحه اصلی نرم افزار کلیک نمائید. کلیه محاسبات را در جدولی مشاهده خواهید کرد که در مورد این مثال، جدول در زیر آمده است.

برنامه نویس: نبی کریمعلی زاده				نرم افزار تسطیح اراضی نسخه اول		
Xc = 196.4516		Yc = 257.4193		Zc = 36.6642		
N=31		Dx=70		Dy=70		R=1.26
Sx(first)= (0.1 , 0.3)				Sy(first)= (0.2 , 0.5)		
sigma(X^2)= 1484700		sigma(Y^2)= 2597000		sigma(X*Y)= 1484700		sigma(X*Z)= 222874.4
						sigma(Y*Z)= 291655.7
Sx* (288309.7) + Sy* (-82983.75) = -410.5781				Sy* (542793.8) + Sx* (-82983.75) = -924.5938		
Sx= -0.001				Sy= -0.002		
CUT (first)=1.99				FILL (first)=2.13		
Nc=13		Nf=18		No Cut/Fill=0		
R(first)=0.9342722				d=0.02		
(x) طول	(y) عرض	(z) ارتفاع اولیه	(z) ارتفاع بعد از تسطیح تصحیح نشده	(dZ) مقدار و نوع عملیات خاکی قبل از تصحیح	(dZ) مقدار و نوع عملیات خاکی بعد از تصحیح	(z) نهایی تصحیح شده
1	1	37	37.17	.17 F	.15 F	37.15
1	2	37.2	37.03	.17 C	.19 C	37.01
1	3	37.29	36.89	.4 C	.42 C	36.87
1	4	37.11	36.75	.36 C	.38 C	36.73
1	5	37.08	36.61	.47 C	.49 C	36.59
1	6	36.48	36.47	.01 C	.03 C	36.45
1	7	36.25	36.33	.08 F	.06 F	36.31
2	1	37.15	37.1	.05 C	.07 C	37.08
2	2	36.95	36.96	.01 F	.01 C	36.94
2	3	36.8	36.82	.02 F	0	36.8
2	4	36.7	36.68	.02 C	.04 C	36.66
2	5	36.71	36.54	.17 C	.19 C	36.52
2	6	36.29	36.4	.11 F	.09 F	36.38
2	7	36.17	36.26	.09 F	.07 F	36.24
3	1	37.02	37.03	.01 F	.01 C	37.01
3	2	36.9	36.89	.01 C	.03 C	36.87
3	3	36.62	36.75	.13 F	.11 F	36.73
3	4	36.7	36.61	.09 C	.11 C	36.59
3	5	36.54	36.47	.07 C	.09 C	36.45
3	6	36.18	36.33	.15 F	.13 F	36.31
3	7	36.02	36.19	.17 F	.15 F	36.17
4	1	36.95	36.96	.01 F	.01 C	36.94
4	2	36.7	36.82	.12 F	.1 F	36.8
4	3	36.45	36.68	.23 F	.21 F	36.66
4	4	36.5	36.54	.04 F	.02 F	36.52
4	5	36.52	36.4	.12 C	.14 C	36.38
5	1	36.65	36.89	.24 F	.22 F	36.87
5	2	36.45	36.75	.3 F	.28 F	36.73
5	3	36.43	36.61	.18 F	.16 F	36.59

5	4	36.4	36.47	.07 F	.05 F	36.45
5	5	36.38	36.33	.05 C	.07 C	36.31
CUT (final)=2.28				FILL (final)=1.8		
Nc (final)=16		Nf (final)=14		No Cut/Fill (final)=1		
R(final)=1.266667						
توجه: واحد مقادیر خروجی برابر واحد مقادیر ورودی شما می باشد.						
http://www.Tastih.Nabi.ir/						

همان طور که در جدول بالا مشاهده میکنید. بخش ابتدایی جدول شامل داده های خام به همراه مقادیر اولیه حاصل از محاسبات می باشد. در جدول آبی رنگ نیز به ترتیب: طول (x)، عرض (y)، ارتفاع اولیه (Z)، ارتفاع بعد از تسطیح تصحیح نشده (Z)، مقدار و نوع عملیات خاکی قبل از تصحیح (dZ)، مقدار و نوع عملیات خاکی بعد از تصحیح (dZ)، ارتفاع نهایی تصحیح شده (Z) آمده است. در انتهای جدول نیز مقادیر نهایی پس از تصحیح محاسبه و آورده شده است.

جهت مشاهده موقعیت و مختصات مرکز ثقل میتوانید از منوی "عملیات" گزینه "نمایش نقطه مرکز ثقل" را کلیک نمائید.

جهت مشاهده حجم کل خاکبرداری و خاکریزی زمین از منوی "عملیات" گزینه "حجم کل خاکبرداری/خاکریزی" را کلیک نمائید.

حجم کل خاکبرداری و خاکریزی منطقه بدین شرح می باشد،
نقاط بین گره ها به روش میان یابی محاسبه شده اند،
واحد ها برابر مکعب واحد مقادیر ورودی می باشند.

حجم خاکبرداری کل: 5792.42
حجم خاکریزی کل: 4179.31

بستن

جهت ذخیره نقشه های ترسیم شده توسط نرم افزار بصورت نقشه بیتی¹ از گزینه "ذخیره نقشه" واقع در منوی "ابزارها" استفاده نمائید.

در انتها، پروژه انجام شده را توسط گزینه "ذخیره سازی" واقع در "منوی اصلی" در قالب یک فایل با پسوند TAS. ذخیره سازی نمائید. فایل تولید شده بدین ترتیب حاوی کلیه داده های مسئله می باشد و قابل جابجایی و بازیابی مجدد توسط گزینه "خواندن پرونده" از "منوی اصلی" این نرم افزار می باشد.

نکات برنامه نویسی نرم افزار

از آنجا که سورس کامل این نرم افزار بیش از 2400 سطر می باشد، توضیح و شرح تک تک سطور در اینجا کمی دور از حوصله و موضوع می باشد. بنابراین به توضیح کلی درمورد شیوه برنامه نویسی می پردازیم.

هسته اصلی این نرم افزار، یک کلاس¹ با نام **clsTastih** می باشد که داده های خام را تحویل گرفته و پس از پردازش داده ها، نتایج محاسبات را به عنوان خروجی به ما تحویل میدهد.

این کلاس طوری برنامه نویسی شده است که علاوه بر استفاده در این نرم افزار به صورت مستقل در برنامه های دیگر قابل استفاده می باشد.

مقادیری که این کلاس به عنوان ورودی دریافت میکند عبارتند از: ارتفاع کلیه نقاط شبکه، حداکثر تعداد نقاط در راستای محور افقی شبکه، حداکثر تعداد نقاط در راستای محور عمودی شبکه، عرض شبکه در جهت محور افقی، عرض شبکه در جهت محور عمودی، مقدار **R**، محدوده مجاز شیب در جهت آبپاری، محدوده مجاز شیب در جهت عمود بر آبپاری. همچنین توسط مقادیر ورودی میتوان تعیین نمود که تصحیح مربوط به شیبها لحاظ گردد یا خیر و دیگر اینکه مختصات مرکز ثقل به صورت خودکار یا دستی محاسبه گردد؛ که در صورتی که مشخص شود مختصات مرکز ثقل باید به صورت دستی اعمال شود، می بایست مختصات طول و عرض آن را نیز به کلاس ارسال نمود.

به مثال زیر توجه کنید:

```
تعریف و ایجاد شیء
Dim Cal As clsTastih
Set Cal = New clsTastih
حداکثر تعداد نقاط در راستای محور افقی شبکه
Cal.in_x0 = 5
حداکثر تعداد نقاط در راستای محور عمودی شبکه
Cal.in_y0 = 5
عرض شبکه در جهت محور افقی
Cal.in_dx = 50
عرض شبکه در جهت محور عمودی
Cal.in_dy = 50
مقدار R
Cal.in_R = 1.32
حداقل محدوده مجاز شیب در جهت آبپاری
Cal.in_Sx1 = 0.1
```

```

' حداکثر محدوده مجاز شیب در جهت آبیاری'
Cal.in_Sx2 = 0.8
' حداقل محدوده مجاز شیب در جهت عمود بر آبیاری'
Cal.in_Sy1 = 0.02
' حداکثر محدوده مجاز شیب در جهت عمود بر آبیاری'
Cal.in_Sy2 = 0.4
' تصحیح شیبها لحاظ شود یا خیر'
Cal.in_CheckShib = True
' طول مختصات نقطه مرکز ثقل'
Cal.in_XcManual = 150
' عرض مختصات نقطه مرکز ثقل'
Cal.in_YcManual = 150
' مختصات مرکز ثقل خودکار حساب شود یا خیر'
Cal.in_CheckCentroid = False
' ارتفاع کلیه نقاط شبکه'
For x = 1 To 5
    For y = 1 To 5
        Cal.in_a(x, y) = Round(Rnd() * 10, 1) + 20 '20~30
    Next y
Next x
' پردازش اطلاعات و انجام محاسبات'
Cal.Run
' ایجاد فایل خروجی حاوی نتایج محاسبات'
Cal.CreateFile ("c:\report.htm")
' نمایش فایل خروجی توسط مرورگر'
Shell "explorer.exe c:\report.htm"

```

در مثالی که از نظر گذشت، دو سطر ابتدای آن جهت ایجاد شیئی¹ از نوع کلاس **Cal** با نام **clstastih** می باشد.

مابقی سطور تا دستور **Cal.Run** به مقدار دهی پارامترهای ورودی شیء ایجاد شده اختصاص دارند. (توجه داشته باشید که کلیه اعداد به صورت فرضی و تصادفی می باشند).

دستور **Cal.Run** عمل پردازش اطلاعات ورودی و انجام محاسبات را انجام خواهد داد.

توسط دستور **Cal.CreateFile ("c:\report.htm")** در مثال فوق کلیه محاسبات انجام شده و خروجی را به صورت منظم و جدول بندی شده در فایل **c:\report.htm** قرار خواهد داد.

نهایتاً توسط دستور **Shell**، فایل تولید شده را به کمک فایل **explorer.exe** اجرا میکنیم و نمایش خواهیم داد.

توضیح هر سطر، قبل از آن نوشته شده است. ضمناً به نام پارامترهای ورودی توجه داشته باشید.

نکته: در مورد نقاطی از شبکه که ارتفاع آن نا معین می باشد، نیازی به مقدار دهی نمی باشد و یا اینکه میتوان مقدار صفر را ارسال نمود.

سورس کامل نرم افزار

سورس کامل نرم افزار بر روی CD ضمیمه موجود می باشد.

محتویات فایل "clsTastih.cls" :

```

'##### Tastih Arazi v1.0 CLASS #####
'##### Programmer: Nabi Karamalizadeh #####
'##### http://www.Tastih.Nabi.ir/ #####
'##### Copyright 2007 All rights reserved. #####

Dim dz() As Single
Const C34 = " "
Const Cr = vbCr

Private Type typOutput
    N As Single
    dX As Single
    dY As Single
    R As Single
    Xc As Single
    Yc As Single
    Zc As Single
    X2 As Single
    Y2 As Single
    XY As Single
    XZ As Single
    YZ As Single
    a1 As Single
    b1 As Single
    c1 As Single
    a2 As Single
    b2 As Single
    c2 As Single
    Sx As Single
    Sy As Single
    Sx1 As Single
    Syl As Single
    Sx2 As Single
    Sy2 As Single
    CUT_first As Single
    FILL_first As Single
    Nc_first As Single
    Nf_first As Single
    NOCF_first As Single
    r_first As Single
    d As Single
    CUT_final As Single
    FILL_final As Single
    Nc_final As Single
    Nf_final As Single
    NOCF_final As Single
    r_final As Single
    dz_string_first() As String
    dz_string_final() As String
    b() As Single
    c() As Single
End Type
Dim OutputData As typOutput

Private Type typInput
    a() As Single
    x0 As Single
    y0 As Single
    dX As Single
    dY As Single
    R As Single
    Sx1 As Single
    Sx2 As Single
    Syl As Single
    Sy2 As Single
    CheckShib As Boolean
    XcManual As Single

```

```

        YcManual As Single
        CheckCentroid As Boolean
    End Type
    Dim InputData As typInput

    Function Run()
    On Error Resume Next

    ReDim OutputData.b(InputData.x0, InputData.y0)
    ReDim OutputData.c(InputData.x0, InputData.y0)
    ReDim dZ(InputData.x0, InputData.y0)
    ReDim OutputData.dZ_string_first(InputData.x0, InputData.y0)
    ReDim OutputData.dZ_string_final(InputData.x0, InputData.y0)

    OutputData.R = InputData.R
    OutputData.dX = InputData.dX
    OutputData.dY = InputData.dY

    For i = 1 To InputData.x0
        For j = 1 To InputData.y0
            If InputData.a(i, j) <> 0 Then N = N + 1
        Next j
    Next i
    OutputData.N = N

    If InputData.CheckCentroid = True Then
        For i = 1 To InputData.y0
            For j = 1 To InputData.x0
                If InputData.a(j, i) <> 0 Then Xc0 = Xc0 + j * InputData.dX
            Next j
        Next i
        Xc = Xc0 / N
        OutputData.Xc = Xc
    Else
        Xc = InputData.XcManual
        OutputData.Xc = InputData.XcManual
    End If

    If InputData.CheckCentroid = True Then
        For i = 1 To InputData.x0
            For j = 1 To InputData.y0
                If InputData.a(i, j) <> 0 Then Yc0 = Yc0 + j * InputData.dY
            Next j
        Next i
        Yc = Yc0 / N
        OutputData.Yc = Yc
    Else
        Yc = InputData.YcManual
        OutputData.Yc = InputData.YcManual
    End If

    For i = 1 To InputData.x0
        For j = 1 To InputData.y0
            If InputData.a(i, j) <> 0 Then Zc0 = Zc0 + InputData.a(i, j)
        Next j
    Next i
    Zc = Zc0 / N
    OutputData.Zc = Zc

    For j = 1 To InputData.y0
        For i = 1 To InputData.x0
            If InputData.a(i, j) <> 0 Then X2 = X2 + (i * InputData.dX) ^ 2
        Next i
    Next j

    For i = 1 To InputData.x0
        For j = 1 To InputData.y0
            If InputData.a(i, j) <> 0 Then Y2 = Y2 + (j * InputData.dY) ^ 2
        Next j
    Next i

    For i = 1 To InputData.x0
        For j = 1 To InputData.y0
            If InputData.a(i, j) <> 0 Then XY = XY + (i * InputData.dX) * (j *
InputData.dY)
        Next j
    Next i

    For i = 1 To InputData.x0
        For j = 1 To InputData.y0
            If InputData.a(i, j) <> 0 Then XZ = XZ + (i * InputData.dX) * InputData.a(i,
j)

```



```

        Next j
    Next i

    For j = 1 To InputData.y0
        For i = 1 To InputData.x0
            If InputData.a(i, j) <> 0 Then YZ = YZ + (j * InputData.dY) * InputData.a(i, j)
        Next i
    Next j

    OutputData.X2 = X2
    OutputData.Y2 = Y2
    OutputData.XY = XY
    OutputData.XZ = XZ
    OutputData.YZ = YZ

    a1 = X2 - N * Xc ^ 2
    b1 = XY - N * Xc * Yc
    c1 = XZ - N * Xc * Zc
    a2 = XY - N * Xc * Yc
    b2 = Y2 - N * Yc ^ 2
    c2 = YZ - N * Yc * Zc
    OutputData.a1 = a1
    OutputData.b1 = b1
    OutputData.c1 = c1
    OutputData.a2 = a2
    OutputData.b2 = b2
    OutputData.c2 = c2

    Sy = (c1 - (a1 * c2 / a2)) / (b1 - (a1 * b2 / a2))
    Sx = (c1 - b1 * y) / a1
    OutputData.Sy = Sy
    OutputData.Sx = Sx

    If InputData.CheckShib = True Then
        OutputData.Sx1 = InputData.Sx1
        OutputData.Sx2 = InputData.Sx2
        OutputData.Sy1 = InputData.Sy1
        OutputData.Sy2 = InputData.Sy2
        Sx1 = InputData.Sx1 / 100
        Sx2 = InputData.Sx2 / 100
        Sy1 = InputData.Sy1 / 100
        Sy2 = InputData.Sy2 / 100
        If Sx < 0 Then Sx1 = -Sx1: Sx2 = -Sx2
        If Sy < 0 Then Sy1 = -Sy1: Sy2 = -Sy2
        dSx1 = Abs(Sx - Sx1)
        dSx2 = Abs(Sx - Sx2)

        dSy1 = Abs(Sy - Sy1)
        dSy2 = Abs(Sy - Sy2)

        If dSx1 < dSx2 Then Sx = Sx1 Else Sx = Sx2
        If dSy1 < dSy2 Then Sy = Sy1 Else Sy = Sy2

        OutputData.Sy = Sy
        OutputData.Sx = Sx
    End If

    ZA0 = Zc - Xc * Sx - Yc * Sy
    For i = 1 To InputData.x0
        For j = 1 To InputData.y0
            If InputData.a(i, j) <> 0 Then
                OutputData.b(i, j) = ZA0 + i * InputData.dX * Sx + j * InputData.dY
            * Sy
                OutputData.b(i, j) = Format(OutputData.b(i, j), "###.###")
            End If
        Next j
    Next i

    For i = 1 To InputData.x0
        For j = 1 To InputData.y0
            If InputData.a(i, j) <> 0 Then
                dZ(i, j) = InputData.a(i, j) - OutputData.b(i, j)
                dZ(i, j) = Round(dZ(i, j), 2)
                If dZ(i, j) > 0 Then CUT = CUT + dZ(i, j): Nc = Nc + 1
                If dZ(i, j) < 0 Then FILL = FILL + dZ(i, j): Nf = Nf + 1
                If dZ(i, j) = 0 Then NOCF = NOCF + 1
            End If
        Next j
    Next i

    OutputData.CUT_first = Format(Abs(CUT), "###.###")
    OutputData.FILL_first = Format(Abs(FILL), "###.###")
    OutputData.r_first = OutputData.CUT_first / OutputData.FILL_first

```

```

OutputData.Nc_first = Nc
OutputData.Nf_first = Nf
OutputData.NOCF_first = NOCF
d = ((1 + InputData.R) / (2 * InputData.R * N)) * (InputData.R * OutputData.FILL_first -
OutputData.CUT_first)
OutputData.d = Format(d, "##.###")

If OutputData.r_first > InputData.R Then d = -d

For i = 1 To InputData.x0
    For j = 1 To InputData.y0
        If InputData.a(i, j) <> 0 Then
            If dZ(i, j) > 0 Then OutputData.dZ_string_first(i, j) =
Format(Abs(dZ(i, j)), "##.###") & " <font color=red>C</font>"
            If dZ(i, j) < 0 Then OutputData.dZ_string_first(i, j) =
Format(Abs(dZ(i, j)), "##.###") & " <font color=blue>F</font>"
            If dZ(i, j) = 0 Then OutputData.dZ_string_first(i, j) = "<font
color=#FF00FF>0</font>"
            dZ(i, j) = dZ(i, j) + d
            dZ(i, j) = Round(dZ(i, j), 2)
            If dZ(i, j) > 0 Then OutputData.dZ_string_final(i, j) =
Format(Abs(dZ(i, j)), "##.###") & " <font color=red>C</font>"
            If dZ(i, j) < 0 Then OutputData.dZ_string_final(i, j) =
Format(Abs(dZ(i, j)), "##.###") & " <font color=blue>F</font>"
            If dZ(i, j) = 0 Then OutputData.dZ_string_final(i, j) = "<font
color=#FF00FF>0</font>"
            OutputData.c(i, j) = InputData.a(i, j) - dZ(i, j)
            OutputData.c(i, j) = Round(OutputData.c(i, j), 2)
        End If
    Next j
Next i
CUT = 0
FILL = 0
Nc = 0
Nf = 0
NOCF = 0
For i = 1 To InputData.x0
    For j = 1 To InputData.y0
        If InputData.a(i, j) <> 0 Then
            If dZ(i, j) > 0 Then CUT = CUT + dZ(i, j): Nc = Nc + 1
            If dZ(i, j) < 0 Then FILL = FILL + dZ(i, j): Nf = Nf + 1
            If dZ(i, j) = 0 Then NOCF = NOCF + 1
        End If
    Next j
Next i
OutputData.CUT_final = Format(Abs(CUT), "##.###")
OutputData.FILL_final = Format(Abs(FILL), "##.###")
OutputData.r_final = OutputData.CUT_final / OutputData.FILL_final
OutputData.Nc_final = Nc
OutputData.Nf_final = Nf
OutputData.NOCF_final = NOCF

Close
End Function

Public Sub CreateFile(FileName As String)

Open FileName For Output As #1

Print #1, "<html>"
Print #1, ""
Print #1, "<head>"
Print #1, "<meta http-equiv=" & Chr(34) & "Content-Type" & Chr(34) & " content=" & Chr(34) &
"text/html; charset=windows-1252" & Chr(34) & ">"
Print #1, "<title>report tastih v1.0</title>"
Print #1, "</head>"
Print #1, "<style type=" & Chr(34) & "text/css" & Chr(34) & ">"
Print #1, "<!--"
Print #1, "td {"
Print #1, "font-family: tahoma, arial;"
Print #1, "font-size: 12px;"
Print #1, "}"
Print #1, "-->"
Print #1, "</style>"
Print #1, "<body>"
Print #1, ""
Print #1, "<table border=" & Chr(34) & "1" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table1" & Chr(34) & ">"
Print #1, " <tr>"
Print #1, " <td>"

```

```

Print #1, "      <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table13" & Chr(34) & " cellspacing=" & Chr(34) & "5" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & " bgcolor=" & Chr(34) & "#000080" & Chr(34) &
">"
Print #1, "      <tr>"
Print #1, "      <td dir=" & Chr(34) & "rtl" & Chr(34) & " width=" & Chr(34) & "50%"
& Chr(34) & "><b><span lang=" & Chr(34) & "fa" & Chr(34) & "><font color=" & Chr(34) &
"FFFFFF" & Chr(34) & ">"
Print #1, "      &#1606;&#1585;&#1605; &#1575;&#1601;&#1586;&#1575;&#1585;
&#1578;&#1587;&#1591;&#1740;&#1581; "
Print #1, "      &#1575;&#1585;&#1575;&#1590;&#1740; &#1606;&#1587;&#1582;&#1607;
&#1575;&#1608;&#1604;</font></span></b></td>"
Print #1, "      <td dir=" & Chr(34) & "rtl" & Chr(34) & " width=" & Chr(34) & "50%"
& Chr(34) & "><b><span lang=" & Chr(34) & "fa" & Chr(34) & "><font color=" & Chr(34) &
"FFFFFF" & Chr(34) & ">&#1576;&#1585;&#1606;&#1575;&#1605;&#1607;
&#1606;&#1608;&#1740;&#1587;: "
Print #1, "      &#1606;&#1576;&#1740; &#1705;&#1585;&#1605;&#1593;&#1604;&#1740;
&#1586;&#1575;&#1583;&#1607;</font></span></b></td>"
Print #1, "      </tr>"
Print #1, "      </table>"
Print #1, "      </td>"
Print #1, "      </tr>"
Print #1, "      <tr>"
Print #1, "      <td>"
Print #1, "      <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table2" & Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & ">"
Print #1, "      <tr>"
Print #1, "      <td>Xc = <b>" & OutputData.Xc & "</b></td>"
Print #1, "      <td>Yc = <b>" & OutputData.Yc & "</b></td>"
Print #1, "      <td>Zc = <b>" & OutputData.Zc & "</b></td>"
Print #1, "      </tr>"
Print #1, "      </table>"
Print #1, "      </td>"
Print #1, "      </tr>"
Print #1, "      <tr>"
Print #1, "      <td>"
Print #1, "      <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table3" & Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & ">"
Print #1, "      <tr>"
Print #1, "      <td>N=<b>" & OutputData.N & "</b></td>"
Print #1, "      <td>Dx=<b>" & OutputData.dX & "</b></td>"
Print #1, "      <td>Dy=<b>" & OutputData.dY & "</b></td>"
Print #1, "      <td>R=<b>" & OutputData.R & "</b></td>"
Print #1, "      </tr>"
Print #1, "      </table>"
Print #1, "      </td>"
Print #1, "      </tr>"
If InputData.CheckShib = True Then
Print #1, "      <tr>"
Print #1, "      <td>"
Print #1, "      <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table4" & Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & ">"
Print #1, "      <tr>"
Print #1, "      <td>Sx(first)= ( <b>" & OutputData.Sx1 & "</b> , <b>" &
OutputData.Sx2 & "</b> )</td>"
Print #1, "      <td>Sy(first)= ( <b>" & OutputData.Sy1 & "</b> , <b>" &
OutputData.Sy2 & "</b> )</td>"
Print #1, "      </tr>"
Print #1, "      </table>"
Print #1, "      </td>"
Print #1, "      </tr>"
End If
Print #1, "      <tr>"
Print #1, "      <td>"
Print #1, "      <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table5" & Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & ">"
Print #1, "      <tr>"
Print #1, "      <td>sigma(X^2)=<b>" & OutputData.X2 & "</b></td>"
Print #1, "      <td>sigma(Y^2)=<b>" & OutputData.Y2 & "</b></td>"
Print #1, "      <td>sigma(X*Y)=<b>" & OutputData.XY & "</b></td>"
Print #1, "      <td>sigma(X*Z)=<b>" & OutputData.XZ & "</b></td>"
Print #1, "      <td>sigma(Y*Z)=<b>" & OutputData.YZ & "</b></td>"
Print #1, "      </tr>"
Print #1, "      </table>"
Print #1, "      </td>"
Print #1, "      </tr>"
Print #1, "      <tr>"
Print #1, "      <td>"

```

```

Print #1, "      <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table6" & Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & ">"
Print #1, "      <tr>"
Print #1, "          <td>Sx* (<b>" & OutputData.a1 & "</b>) + Sy* (<b>" & OutputData.b1 &
"</b>) = <b>" & OutputData.c1 & "</b></td>"
Print #1, "          <td>Sy* (<b>" & OutputData.b2 & "</b>) + Sx* (<b>" & OutputData.a2 &
"</b>) = <b>" & OutputData.c2 & "</b></td>"
Print #1, "      </tr>"
Print #1, "      </table>"
Print #1, "      </td>"
Print #1, " </tr>"
Print #1, " <tr>"
Print #1, "     <td>"
Print #1, "     <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table7" & Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & ">"
Print #1, "     <tr>"
Print #1, "         <td>Sx=<b>" & OutputData.Sx & "</b></td>"
Print #1, "         <td>Sy=<b>" & OutputData.Sy & "</b></td>"
Print #1, "     </tr>"
Print #1, "     </table>"
Print #1, "     </td>"
Print #1, " </tr>"
Print #1, " <tr>"
Print #1, "     <td>"
Print #1, "     <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table8" & Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & ">"
Print #1, "     <tr>"
Print #1, "         <td>CUT (first)=<b>" & OutputData.CUT_first & "</b></td>"
Print #1, "         <td>FILL (first)=<b>" & OutputData.FILL_first & "</b></td>"
Print #1, "     </tr>"
Print #1, "     </table>"
Print #1, "     </td>"
Print #1, " </tr>"
Print #1, " <tr>"
Print #1, "     <td>"
Print #1, "     <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table9" & Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & ">"
Print #1, "     <tr>"
Print #1, "         <td>Nc=<b>" & OutputData.Nc_first & "</b></td>"
Print #1, "         <td>Nf=<b>" & OutputData.Nf_first & "</b></td>"
Print #1, "         <td>No Cut/Fill=<b>" & OutputData.NOCF_first & "</b></td>"
Print #1, "     </tr>"
Print #1, "     </table>"
Print #1, "     </td>"
Print #1, " </tr>"
Print #1, " <tr>"
Print #1, "     <td>"
Print #1, "     <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table9" & Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & ">"
Print #1, "     <tr>"
Print #1, "         <td>R(first)=<b>" & OutputData.r_first & "</b></td>"
Print #1, "         <td>d=<b>" & OutputData.d & "</b></td>"
Print #1, "     </tr>"
Print #1, "     </table>"
Print #1, "     </td>"
Print #1, " </tr>"
Print #1, " <tr>"
Print #1, "     <td>"
Print #1, "     <table border=" & Chr(34) & "1" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table12" & Chr(34) & " bordercolor=" & Chr(34) & "#000000" &
Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34) & " style=" & Chr(34) & "border-
collapse: collapse" & Chr(34) & " cellpadding=" & Chr(34) & "2" & Chr(34) & ">"
Print #1, "     <tr>"
Print #1, "         <td bgcolor=" & Chr(34) & "#000080" & Chr(34) & " align=" & Chr(34)
& "center" & Chr(34) & "><b><font color=" & Chr(34) & "#FFFFFF" & Chr(34) & ">(x)
&#1591;&#1608;&#1604;</font></b></td>"
Print #1, "         <td bgcolor=" & Chr(34) & "#000080" & Chr(34) & " align=" & Chr(34)
& "center" & Chr(34) & "><b><font color=" & Chr(34) & "#FFFFFF" & Chr(34) & ">(y)
&#1593;&#1585;&#1590;</font></b></td>"
Print #1, "         <td bgcolor=" & Chr(34) & "#000080" & Chr(34) & " align=" & Chr(34)
& "center" & Chr(34) & "><b><font color=" & Chr(34) & "#FFFFFF" & Chr(34) & ">(z)
&#1575;&#1585;&#1578;&#1601;&#1575;&#1593;
&#1575;&#1608;&#1604;&#1610;&#1607;</font></b></td>"
Print #1, "         <td bgcolor=" & Chr(34) & "#000080" & Chr(34) & " align=" & Chr(34)
& "center" & Chr(34) & "><b><font color=" & Chr(34) & "#FFFFFF" & Chr(34) & ">(z)
&#1575;&#1585;&#1578;&#1601;&#1575;&#1593; &#1576;&#1593;&#1583; &#1575;&#1586;

```

```

&#1578;&#1587;&#1591;&#1610;&#1581; &#1578;&#1589;&#1581;&#1610;&#1581;
&#1606;&#1588;&#1583;&#1607;</font></b></td>"
Print #1, "
    <td bgcolor=" & Chr(34) & "#000080" & Chr(34) & " align=" & Chr(34)
& "center" & Chr(34) & "><b><font color=" & Chr(34) & "#FFFFFF" & Chr(34) & ">(dZ)
&#1605;&#1602;&#1583;&#1575;&#1585; &#1608; &#1606;&#1608;&#1593;
&#1593;&#1605;&#1604;&#1610;&#1575;&#1578; &#1582;&#1575;&#1705;&#1740;
&#1602;&#1576;&#1604; &#1575;&#1586; &#1578;&#1589;&#1581;&#1610;&#1581;</font></b></td>"
Print #1, "
    <td bgcolor=" & Chr(34) & "#000080" & Chr(34) & " align=" & Chr(34)
& "center" & Chr(34) & "><b><font color=" & Chr(34) & "#FFFFFF" & Chr(34) & ">(dZ)
&#1605;&#1602;&#1583;&#1575;&#1585; &#1608; &#1606;&#1608;&#1593;
&#1593;&#1605;&#1604;&#1610;&#1575;&#1578; &#1582;&#1575;&#1705;&#1740;
&#1576;&#1593;&#1583; &#1575;&#1586; &#1578;&#1589;&#1581;&#1610;&#1581;</font></b></td>"
Print #1, "
    <td bgcolor=" & Chr(34) & "#000080" & Chr(34) & " align=" & Chr(34)
& "center" & Chr(34) & "><b><font color=" & Chr(34) & "#FFFFFF" & Chr(34) & ">(z)
&#1575;&#1585;&#1578;&#1601;&#1575;&#1593; &#1606;&#1607;&#1575;&#1740;&#1740;
&#1578;&#1589;&#1581;&#1740;&#1581; &#1588;&#1583;&#1607;</font></b></td>"
Print #1, "
</tr>"
color1 = "#F0F0FF"
color2 = "#F4F4FF"
For i = 1 To InputData.x0
    For j = 1 To InputData.y0
        If InputData.a(i, j) <> 0 Then
            If color_flag = False Then
                color0 = color1
                color_flag = True
            Else
                color0 = color2
                color_flag = False
            End If
Print #1, "
Print #1, "
    <td align=" & Chr(34) & "center" & Chr(34) & " bgcolor=" & Chr(34) &
color0 & Chr(34) & "><b>" & i & "</b></td>"
Print #1, "
    <td align=" & Chr(34) & "center" & Chr(34) & " bgcolor=" & Chr(34) &
color0 & Chr(34) & "><b>" & j & "</b></td>"
Print #1, "
    <td align=" & Chr(34) & "center" & Chr(34) & " bgcolor=" & Chr(34) &
color0 & Chr(34) & "><b>" & InputData.a(i, j) & "</b></td>"
Print #1, "
    <td align=" & Chr(34) & "center" & Chr(34) & " bgcolor=" & Chr(34) &
color0 & Chr(34) & "><b>" & OutputData.b(i, j) & "</b></td>"
Print #1, "
    <td align=" & Chr(34) & "center" & Chr(34) & " bgcolor=" & Chr(34) &
color0 & Chr(34) & "><b>" & OutputData.dZ_string_first(i, j) & "</b></td>"
Print #1, "
    <td align=" & Chr(34) & "center" & Chr(34) & " bgcolor=" & Chr(34) &
color0 & Chr(34) & "><b>" & OutputData.dZ_string_final(i, j) & "</b></td>"
Print #1, "
    <td align=" & Chr(34) & "center" & Chr(34) & " bgcolor=" & Chr(34) &
color0 & Chr(34) & "><b>" & OutputData.c(i, j) & "</b></td>"
Print #1, "
    </tr>"
        End If
    Next j
Next i

Print #1, "
Print #1, "
Print #1, " </tr>"
Print #1, " <tr>"
Print #1, "
    <td>"
Print #1, "
    <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table10" & Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & ">"
Print #1, "
    <tr>"
Print #1, "
        <td>CUT (final)=<b>" & OutputData.CUT_final & "</b></td>"
Print #1, "
        <td>FILL (final)=<b>" & OutputData.FILL_final & "</b></td>"
Print #1, "
    </tr>"
Print #1, "
    </table>"
Print #1, "
    </td>"
Print #1, " </tr>"
Print #1, " <tr>"
Print #1, "
    <td>"
Print #1, "
    <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table11" & Chr(34) & " cellspacing=" & Chr(34) & "0" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & ">"
Print #1, "
    <tr>"
Print #1, "
        <td>Nc (final)=<b>" & OutputData.Nc_final & "</b></td>"
Print #1, "
        <td>Nf (final)=<b>" & OutputData.Nf_final & "</b></td>"
Print #1, "
        <td>No Cut/Fill (final)=<b>" & OutputData.NOCF_final & "</b></td>"
Print #1, "
    </tr>"
Print #1, "
    </table>"
Print #1, "
    </td>"
Print #1, " </tr>"
Print #1, " <tr>"
Print #1, "
    <td>R(final)=<b>" & OutputData.r_final & "</b></td>"
Print #1, "
    </tr>"
Print #1, " <tr>"

```

```

Print #1, "          <td><p align=" & Chr(34) & "right" & Chr(34) & " dir=" & Chr(34) & "rtl" &
Chr(34) & "><b>#1578;#1608;#1580;#1607;: </b>#1608;#1575;#1581;#1583;
&#1605;#1602;#1575;#1583;#1740;#1585; &#1582;#1585;#1608;#1580;#1740;
&#1576;#1585;#1575;#1576;#1585; &#1608;#1575;#1581;#1583;
&#1605;#1602;#1575;#1583;#1740;#1585; &#1608;#1585;#1608;#1583;#1740;
&#1588;#1605;#1575; &#1605;#1740; &#1576;#1575;#1588;#1583;.</td>"
Print #1, " </tr>"
Print #1, " <tr>"
Print #1, "          <td>"
Print #1, "          <table border=" & Chr(34) & "0" & Chr(34) & " width=" & Chr(34) & "100%" &
Chr(34) & " id=" & Chr(34) & "table14" & Chr(34) & " cellspacing=" & Chr(34) & "5" & Chr(34)
& " cellpadding=" & Chr(34) & "0" & Chr(34) & " bgcolor=" & Chr(34) & "#000080" & Chr(34) &
">"
Print #1, "          <tr>"
Print #1, "          <td>"
Print #1, "          <p align=" & Chr(34) & "center" & Chr(34) & ">"
Print #1, "          <a target=" & Chr(34) & "_blank" & Chr(34) & " href=" & Chr(34) & "
http://www.Tastih.Nabi.ir/" & Chr(34) & ">"
Print #1, "          <font color=" & Chr(34) & "#FFFFFF" & Chr(34) & ">"
Print #1, "          http://www.Tastih.Nabi.ir/</font></a></td>"
Print #1, "          </tr>"
Print #1, "          </table>"
Print #1, "          </td>"
Print #1, " </tr>"
Print #1, " </table>"
Print #1, " "
Print #1, " </body>"
Print #1, " "
Print #1, " </html>"

Close #1
End Sub

Property Get in_X0() As Single
    in_X0 = InputData.x0
End Property

Property Let in_X0(tx0 As Single)
    InputData.x0 = tx0
End Property

Property Get in_Y0() As Single
    in_Y0 = InputData.y0
End Property

Property Let in_Y0(ty0 As Single)
    InputData.y0 = ty0
End Property

Property Get in_dX() As Single
    in_dX = InputData.dX
End Property

Property Let in_dX(tdX As Single)
    InputData.dX = tdX
End Property

Property Get in_dY() As Single
    in_dY = InputData.dY
End Property

Property Let in_dY(tdY As Single)
    InputData.dY = tdY
End Property

Property Get in_R() As Single
    in_R = InputData.R
End Property

Property Let in_R(tR As Single)
    InputData.R = tR
End Property

Property Get in_Sx1() As Single
    in_Sx1 = InputData.Sx1
End Property

Property Let in_Sx1(tSx1 As Single)
    InputData.Sx1 = tSx1
End Property

Property Get in_Sx2() As Single

```

```

        in_Sx2 = InputData.Sx2
    End Property

    Property Let in_Sx2(tSx2 As Single)
        InputData.Sx2 = tSx2
    End Property

    Property Get in_Sy1() As Single
        in_Sy1 = InputData.Sy1
    End Property

    Property Let in_Sy1(tSy1 As Single)
        InputData.Sy1 = tSy1
    End Property

    Property Get in_Sy2() As Single
        in_Sy2 = InputData.Sy2
    End Property

    Property Let in_Sy2(tSy2 As Single)
        InputData.Sy2 = tSy2
    End Property

    Property Get in_CheckShib() As Boolean
        in_CheckShib = InputData.CheckShib
    End Property

    Property Let in_CheckShib(tCheckShib As Boolean)
        InputData.CheckShib = tCheckShib
    End Property

    Property Get in_XcManual() As Single
        in_XcManual = InputData.XcManual
    End Property

    Property Let in_XcManual(tXcManual As Single)
        InputData.XcManual = tXcManual
    End Property

    Property Get in_YcManual() As Single
        in_YcManual = InputData.YcManual
    End Property

    Property Let in_YcManual(tYcManual As Single)
        InputData.YcManual = tYcManual
    End Property

    Property Get in_CheckCentroid() As Boolean
        in_CheckCentroid = InputData.CheckCentroid
    End Property

    Property Let in_CheckCentroid(tCheckCentroid As Boolean)
        InputData.CheckCentroid = tCheckCentroid
    End Property

    Public Property Get in_a(ByVal Index1 As Integer, _
        ByVal Index2 As Integer) As Single

        If Index1 <= UBound(InputData.a, 1) And _
            Index2 <= UBound(InputData.a, 2) Then in_a = InputData.a(Index1, Index2)

    End Property

    Public Property Let in_a(ByVal Index1 As Integer, _
        ByVal Index2 As Integer, _
        NewNumber As Single)

        Dim mArray() As Single
        mArray = InputData.a

        If Index1 > UBound(InputData.a, 1) Then ReDim InputData.a(Index1, UBound(InputData.a, 2))
        If Index2 > UBound(InputData.a, 2) Then ReDim InputData.a(UBound(InputData.a, 1), Index2)

        For ISum1 = LBound(mArray, 1) To UBound(mArray, 1)
            For ISum2 = LBound(mArray, 2) To UBound(mArray, 2)
                InputData.a(ISum1, ISum2) = mArray(ISum1, ISum2)
            Next ISum2
        Next ISum1

        InputData.a(Index1, Index2) = NewNumber

    End Property

```

```
Private Sub Class_Initialize()
    ReDim InputData.a(0, 0)
End Sub

Function out_Xc() As Single
    out_Xc = OutputData.Xc
End Function

Function out_Yc() As Single
    out_Yc = OutputData.Yc
End Function

Property Get CutOrFill(x As Long, y As Long) As Single
    On Error Resume Next
    CutOrFill = dz(x, y)
End Property

'##### Tastih Arazi v1.0 CLASS #####
'##### Programmer: Nabi Karamalizadeh #####
'##### http://www.Tastih.Nabi.ir/ #####
'##### Copyright 2007 All rights reserved. #####
```

محتویات فایل "frmMain.frm":

```
Private Declare Function GetCapture Lib "user32" () As Long
Private Declare Function SetCapture Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function ReleaseCapture Lib "user32" () As Long

Dim rs As ADOR.Recordset

Dim First_Load As Boolean
Dim First_Load2 As Boolean

Dim StepH As Single
Dim StepW As Single
Dim BaseH As Single
Dim BaseW As Single

Dim b1 As Boolean
Dim px As Single
Dim py As Single

Dim NodeX As Single
Dim NodeY As Single

Dim MaxW As Single
Dim MaxH As Single
Dim h() As Single

Function DaroonYabi(a As Single, b As Single, d As Single, x As Single) As Single
    'a : ertefa noghte shoro
    'b : ertefa noghte payan
    'd : fasele do noghte
    'x : fasele noghte majhool
    'DaroonYabi : ertefa noghte majhool
    DaroonYabi = (-x * (a - b) + d * a) / d
End Function

Private Sub MoveMouseCapture(ByVal x As Single, ByVal y As Single)
    If GetCapture() <> Earth.hwnd Then
        If GetCapture() = Earth.hwnd Then Exit Sub
        SetCapture Earth.hwnd
    End If
    If x < 0 Or x > Earth.Width Or y < 0 Or y > Earth.Height Then
        ReleaseCapture
    End If
End Sub

Sub Earth_Refresh(Optional Flag_CLS As Boolean = True)
    On Error Resume Next
    Me.MousePointer = 11
    PBar1.Visible = True
    PBar1.Value = 0
    PBar1.Min = 0
    PBar1.Max = 1
```



```

PBar1.Max = PBar1.Max + 3
PBar1.Max = PBar1.Max + ((Earth.ScaleHeight - BaseH) \ StepH + 1)
PBar1.Max = PBar1.Max + ((Earth.ScaleWidth - BaseW) \ StepW + 1)
PBar1.Max = PBar1.Max + 2 * (((Earth.ScaleWidth - BaseW) \ StepW + 1) * ((Earth.ScaleHeight - BaseH) \ StepH + 1))
PBar1.Max = PBar1.Max + rs.RecordCount

Dim i As Long
If StepH = 0 Then StepH = 1
If StepW = 0 Then StepW = 1

If Flag_CLS = True Then Earth.Cls

PBar1.Value = PBar1.Value + 1

'Horizon
For i = BaseH To Earth.ScaleHeight Step StepH
    Earth.Line (0, i)-(Earth.ScaleWidth, i)
    PBar1.Value = PBar1.Value + 1
Next i

'Vertical
For i = BaseW To Earth.ScaleWidth Step StepW
    Earth.Line (i, 0)-(i, Earth.ScaleHeight)
    PBar1.Value = PBar1.Value + 1
Next i

For i = BaseW To Earth.Width Step StepW
    For j = BaseH To Earth.Height Step StepH
        Earth.DrawWidth = 1
        Earth.PSet (i, j)
        Earth.Circle (i, j), 4
        If Flag_CLS = True Then
            Earth.Print " (" & (i - BaseW) / StepW + 1 & "," & (j - BaseH) / StepH + 1 & ")"
        End If
        Earth.DrawMode = 13
        MaxW = (i - BaseW) / StepW + 1
        MaxH = (j - BaseH) / StepH + 1
        PBar1.Value = PBar1.Value + 1
    Next j
Next i

Call H_Refresh

PBar1.Value = PBar1.Value + 1

rs.MoveFirst
While Not rs.EOF
    rs.Delete
    rs.MoveNext
    PBar1.Value = PBar1.Value + 1
Wend

PBar1.Value = PBar1.Value + 1
For i = BaseW To Earth.ScaleWidth Step StepW
    For j = BaseH To Earth.ScaleHeight Step StepH
        rs.AddNew
        rs.Fields(0).Value = (i - BaseW) / StepW + 1
        rs.Fields(1).Value = (j - BaseH) / StepH + 1
        rs.Fields(2).Value = h(rs.Fields(0).Value, rs.Fields(1).Value)
        PBar1.Value = PBar1.Value + 1
    Next j
Next i
rs.MoveFirst

ScrollBaseH.Max = Earth.ScaleHeight - StepH
ScrollStepH.Max = Earth.ScaleHeight - BaseH

ScrollBaseW.Max = Earth.ScaleWidth - StepW
ScrollStepW.Max = Earth.ScaleWidth - BaseW
PBar1.Value = PBar1.Value + 1
PBar1.Visible = False
Me.MousePointer = 1
End Sub

Function Scroll_Refresh()
    ScrollW.Width = 20
    ScrollH.Height = 20

    ScrollW.Top = 0
    ScrollW.Left = Back.ScaleWidth - ScrollW.Width

```

```

ScrollW.Height = Back.ScaleHeight - ScrollH.Height

ScrollH.Top = Back.ScaleHeight - ScrollH.Height
ScrollH.Left = 0
ScrollH.Width = Back.ScaleWidth - ScrollW.Width

picTemp.Width = ScrollW.Width
picTemp.Height = ScrollH.Height
picTemp.Left = Back.ScaleWidth - picTemp.Width
picTemp.Top = Back.ScaleHeight - picTemp.Height

    ScrollW.Value = 0
    If Earth.Height > Back.Height - ScrollH.Height Then
        ScrollW.Enabled = True
        ScrollW.Min = 0
        ScrollW.Max = Earth.Height - Back.Height + ScrollH.Height
    Else
        ScrollW.Enabled = False
    End If
    ScrollH.Value = 0
    If Earth.Width > Back.Width - ScrollW.Width Then
        ScrollH.Enabled = True
        ScrollH.Min = 0
        ScrollH.Max = Earth.Width - Back.Width + ScrollW.Width
        If PicBold = False Then ScrollH.SmallChange = 8 * 15 Else If PicBold = True Then
ScrollH.SmallChange = 9 * 15
        Else
            ScrollH.Enabled = False
        End If
    End If
End Function

Private Sub chkCentroid_Click()
If chkCentroid.Value = 1 Then
    txtXcManual.Enabled = False: txtXcManual.BackColor = RGB(236, 233, 216)
    txtYcManual.Enabled = False: txtYcManual.BackColor = RGB(236, 233, 216)
Else
    txtXcManual.Enabled = True: txtXcManual.BackColor = RGB(255, 255, 255)
    txtYcManual.Enabled = True: txtYcManual.BackColor = RGB(255, 255, 255)
End If
End Sub

Private Sub chkColor_Click(Index As Integer)
If chkColor(Index).Value = 1 Then
    txtColor(Index).Enabled = True
    cmdColor(Index).Visible = True
Else
    txtColor(Index).Enabled = False
    cmdColor(Index).Visible = False
End If
End Sub

Private Sub chkShib_Click()
If chkShib.Value = 1 Then
    txtSx1.Enabled = True: txtSx1.BackColor = RGB(255, 255, 255)
    txtSx2.Enabled = True: txtSx2.BackColor = RGB(255, 255, 255)
    txtSy1.Enabled = True: txtSy1.BackColor = RGB(255, 255, 255)
    txtSy2.Enabled = True: txtSy2.BackColor = RGB(255, 255, 255)
Else
    txtSx1.Enabled = False: txtSx1.BackColor = RGB(236, 233, 216)
    txtSx2.Enabled = False: txtSx2.BackColor = RGB(236, 233, 216)
    txtSy1.Enabled = False: txtSy1.BackColor = RGB(236, 233, 216)
    txtSy2.Enabled = False: txtSy2.BackColor = RGB(236, 233, 216)
End If
End Sub

Private Sub cmdColor_Click(Index As Integer)
On Error GoTo errlabel
    dlgColor.CancelError = True
    dlgColor.ShowColor
    cmdColor(Index).BackColor = dlgColor.Color
    cmdColor(Index).Caption = "#" & TaghirMabna(dlgColor.Color)
Exit Sub
errlabel:
End Sub

Private Sub cmdTastihMap_Click()
Dim xx1 As Single, yy1 As Single, h1 As Single, h2 As Single
Dim X1 As Long, Y1 As Long, X2 As Long, Y2 As Long, X3 As Long, Y3 As Long, X4 As Long, Y4
As Long, i As Long, j As Long
Dim ok As Boolean

```

```

Me.MousePointer = 11

Dim Cal As clsTastih
Set Cal = New clsTastih

Cal.in_X0 = MaxW
Cal.in_Y0 = MaxH
Cal.in_dX = StepW
Cal.in_dY = StepH
Cal.in_R = Val(txtR.Text)
Cal.in_Sx1 = Val(txtSx1.Text)
Cal.in_Sx2 = Val(txtSx2.Text)
Cal.in_Sy1 = Val(txtSy1.Text)
Cal.in_Sy2 = Val(txtSy2.Text)
Cal.in_CheckShib = chkShib.Value
Cal.in_XcManual = Val(txtXcManual.Text)
Cal.in_YcManual = Val(txtYcManual.Text)
Cal.in_CheckCentroid = chkCentroid.Value
rs.MoveFirst
While Not rs.EOF
    If Val(rs.Fields(2).Value) <> 0 Then ok = True
    Cal.in_a(rs.Fields(0).Value, rs.Fields(1).Value) = Val(rs.Fields(2).Value)
    rs.MoveNext
Wend
If ok = False Then MsgBox "ایده نکرده وارد را نقاط از هیچیک ارتفاع", vbInformation,
"Warning": rs.MoveFirst: tabOption.Tabs(2).Selected = True: Me.MousePointer = 0: Exit Sub
Cal.Run

PBar1.Visible = True

Earth.Cls

Call H_Refresh

For i = BaseW To (MaxW - 1) * StepW + BaseW - 1
    For j = BaseH To (MaxH - 1) * StepH + BaseH - 1

        X1 = (((i - BaseW) \ StepW) * StepW + BaseW) - BaseW / StepW + 1
        Y1 = (((j - BaseH) \ StepH) * StepH + BaseH) - BaseH / StepH + 1

        X2 = (((i - BaseW) \ StepW) * StepW + BaseW + StepW) - BaseW / StepW + 1
        Y2 = (((j - BaseH) \ StepH) * StepH + BaseH) - BaseH / StepH + 1

        X3 = (((i - BaseW) \ StepW) * StepW + BaseW) - BaseW / StepW + 1
        Y3 = (((j - BaseH) \ StepH) * StepH + BaseH + StepH) - BaseH / StepH + 1

        X4 = (((i - BaseW) \ StepW) * StepW + BaseW + StepW) - BaseW / StepW + 1
        Y4 = (((j - BaseH) \ StepH) * StepH + BaseH + StepH) - BaseH / StepH + 1

        xx1 = i - ((X1 - 1) * StepW + BaseW)
        yy1 = j - ((Y1 - 1) * StepH + BaseH)

        If h(X1, Y1) <> 0 And h(X2, Y2) <> 0 And h(X3, Y3) <> 0 And h(X4, Y4) <> 0 Then

            h1 = DaroonYabi(Cal.CutOrFill(X1, Y1), Cal.CutOrFill(X3, Y3), StepH, yy1)
            h2 = DaroonYabi(Cal.CutOrFill(X2, Y2), Cal.CutOrFill(X4, Y4), StepH, yy1)
            h3 = DaroonYabi(h1, h2, StepW, xx1)
            If h3 > 0 Then Earth.PSet (i, j), RGB(255, 0, 0)
            If h3 < 0 Then Earth.PSet (i, j), RGB(0, 0, 255)
            If h3 = 0 Then Earth.PSet (i, j), RGB(0, 0, 0)

        End If

    Next j
Next i

For i = 1 To MaxW
    For j = 1 To MaxH
        Earth.DrawMode = 3
        Earth.PSet ((i - 1) * StepW + BaseW, (j - 1) * StepH + BaseH)
        Earth.DrawMode = 13
        Earth.ForeColor = RGB(0, 0, 0)
        If h(i, j) <> 0 Then
            If Cal.CutOrFill(i, j) > 0 Then Earth.Print Abs(Cal.CutOrFill(i, j)) & " C"
            If Cal.CutOrFill(i, j) < 0 Then Earth.Print Abs(Cal.CutOrFill(i, j)) & " F"
            If Cal.CutOrFill(i, j) = 0 Then Earth.Print "0"
        End If
    Next j
Next i

Call Earth_Refresh(False)

```

```

PBar1.Visible = False
Me.MousePointer = 0

End Sub

Private Sub cmdTopo_Click()
Dim xx1 As Single, yy1 As Single, h1 As Single, h2 As Single

Me.MousePointer = 11

rs.MoveFirst
While Not rs.EOF
    If Val(rs.Fields(2).Value) <> 0 Then GoTo exitlabel
    rs.MoveNext
Wend
MsgBox "اید نکرده وارد را نقاط از هیچک ارتفاع", vbInformation, "Warning": rs.MoveFirst:
tabOption.Tabs(2).Selected = True: Me.MousePointer = 0: Exit Sub
exitlabel:

PBar1.Visible = True

Earth.Cls

Call H_Refresh

Dim ColorH(10)
Dim ColorC(10)
p = 0
For i = 0 To 8
    If chkColor(i).Value = 1 Then
        p = p + 1
        ColorH(p) = Val(txtColor(i).Text)
        ColorC(p) = cmdColor(i).BackColor
    End If
Next i

For i = 1 To p - 1
    For j = i + 1 To p
        If ColorH(i) > ColorH(j) Then
            temp = ColorH(i): ColorH(i) = ColorH(j): ColorH(j) = temp
            temp = ColorC(i): ColorC(i) = ColorC(j): ColorC(j) = temp
        End If
    Next j
Next i

For i = BaseW To (MaxW - 1) * StepW + BaseW - 1
    For j = BaseH To (MaxH - 1) * StepH + BaseH - 1

        X1 = (((i - BaseW) \ StepW) * StepW + BaseW) - BaseW / StepW + 1
        Y1 = (((j - BaseH) \ StepH) * StepH + BaseH) - BaseH / StepH + 1

        X2 = (((i - BaseW) \ StepW) * StepW + BaseW + StepW) - BaseW / StepW + 1
        Y2 = (((j - BaseH) \ StepH) * StepH + BaseH) - BaseH / StepH + 1

        X3 = (((i - BaseW) \ StepW) * StepW + BaseW) - BaseW / StepW + 1
        Y3 = (((j - BaseH) \ StepH) * StepH + BaseH + StepH) - BaseH / StepH + 1

        X4 = (((i - BaseW) \ StepW) * StepW + BaseW + StepW) - BaseW / StepW + 1
        Y4 = (((j - BaseH) \ StepH) * StepH + BaseH + StepH) - BaseH / StepH + 1

        xx1 = i - ((X1 - 1) * StepW + BaseW)
        yy1 = j - ((Y1 - 1) * StepH + BaseH)

        If h(X1, Y1) <> 0 And h(X2, Y2) <> 0 And h(X3, Y3) <> 0 And h(X4, Y4) <> 0 Then

            h1 = DaroonYabi(h(X1, Y1), h(X3, Y3), StepH, yy1)
            h2 = DaroonYabi(h(X2, Y2), h(X4, Y4), StepH, yy1)
            h3 = DaroonYabi(h1, h2, StepW, xx1)

            For k = 1 To p
                If h3 < ColorH(k) Then Earth.PSet (i, j), ColorC(k): GoTo breaklabel
            Next k
            Earth.PSet (i, j), RGB(255, 255, 255)
breaklabel:

            End If

        Next j
    Next i

```

```

For i = 1 To MaxW
    For j = 1 To MaxH
        Earth.DrawMode = 3
        Earth.PSet ((i - 1) * StepW + BaseW, (j - 1) * StepH + BaseH)
        Earth.DrawMode = 13
        If h(i, j) <> 0 Then
            Earth.Print h(i, j)
        End If
    Next j
Next i

Call Earth_Refresh(False)

PBar1.Visible = False
Me.MousePointer = 0

End Sub

Private Sub Command1_Click()
chkShib.Value = 0
End Sub

Private Sub cmdCal_Click()
Dim ok As Boolean
Me.MousePointer = 11

Dim Cal As clsTastih
Set Cal = New clsTastih

Cal.in_X0 = MaxW
Cal.in_Y0 = MaxH
Cal.in_dX = StepW
Cal.in_dY = StepH
Cal.in_R = Val(txtR.Text)
Cal.in_Sx1 = Val(txtSx1.Text)
Cal.in_Sx2 = Val(txtSx2.Text)
Cal.in_Sy1 = Val(txtSy1.Text)
Cal.in_Sy2 = Val(txtSy2.Text)
Cal.in_CheckShib = chkShib.Value
Cal.in_XcManual = Val(txtXcManual.Text)
Cal.in_YcManual = Val(txtYcManual.Text)
Cal.in_CheckCentroid = chkCentroid.Value

rs.MoveFirst
While Not rs.EOF
    If Val(rs.Fields(2).Value) <> 0 Then ok = True
    Cal.in_a(rs.Fields(0).Value, rs.Fields(1).Value) = Val(rs.Fields(2).Value)
    rs.MoveNext
Wend
If ok = False Then MsgBox "اید نکرده وارد را نقاط از هیچیک ارتفاع", vbInformation,
"Warning": rs.MoveFirst: tabOption.Tabs(2).Selected = True: Me.MousePointer = 0: Exit Sub

Cal.Run

Cal.CreateFile (FILE_REPORT_TEMP)

frmRepView.WebBrowser1.Navigate (FILE_REPORT_TEMP)
frmRepView.Show 1
Me.MousePointer = 0
End Sub

Private Sub Earth_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)
If NodeX > 0 And NodeY > 0 Then
    rs.MoveFirst
    While Not rs.EOF
        If NodeX = rs.Fields(0).Value And NodeY = rs.Fields(1).Value Then GoTo lblExit
        rs.MoveNext
    Wend
lblExit:
    tabOption.Tabs(2).Selected = True
    DataGrid1.SetFocus
    Exit Sub
End If

Earth.MousePointer = 99

```

```

Earth.MouseIcon = imgHandGet2.Picture
px = x
py = y
b1 = True

End Sub

Private Sub Earth_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)

If b1 = True Then
    Earth.MousePointer = 99
    Earth.MouseIcon = imgHandGet2.Picture
    If Earth.Width > Back.Width - ScrollW.Width Then
        If px > x Then
            Earth.Left = Earth.Left - Abs(px - x)
            If Earth.Left < Back.Width - Earth.Width - ScrollW.Width Then Earth.Left =
Back.Width - Earth.Width - ScrollW.Width: px = x
            ScrollH.Value = -Earth.Left
        End If
        If px < x Then
            Earth.Left = Earth.Left + Abs(px - x)
            If Earth.Left > 0 Then Earth.Left = 0: px = x
            ScrollH.Value = -Earth.Left
        End If
    End If
    If Earth.Height > Back.Height - ScrollH.Height Then
        If py < y Then
            Earth.Top = Earth.Top + Abs(py - y)
            If Earth.Top > 0 Then Earth.Top = 0: py = y
            ScrollW.Value = -Earth.Top
        End If
        If py > y Then
            Earth.Top = Earth.Top - Abs(py - y)
            If Earth.Top < Back.Height - Earth.Height - ScrollH.Height Then Earth.Top =
Back.Height - Earth.Height - ScrollH.Height: py = y
            ScrollW.Value = -Earth.Top
        End If
    End If
Exit Sub
End If

Node.Visible = False
StatusBar1.Panels.Item(1).Text = ""

Earth.MousePointer = 99
Earth.MouseIcon = imgHandGet1.Picture

NodeX = 0
NodeY = 0

Call MoveMouseCapture(x, y)

x_now = x
y_now = y

X1 = ((x - BaseW) \ StepW) * StepW + BaseW
Y1 = ((y - BaseH) \ StepH) * StepH + BaseH

X2 = ((x - BaseW) \ StepW) * StepW + BaseW + StepW
Y2 = ((y - BaseH) \ StepH) * StepH + BaseH

X3 = ((x - BaseW) \ StepW) * StepW + BaseW
Y3 = ((y - BaseH) \ StepH) * StepH + BaseH + StepH

X4 = ((x - BaseW) \ StepW) * StepW + BaseW + StepW
Y4 = ((y - BaseH) \ StepH) * StepH + BaseH + StepH

d1 = Sqr((x_now - X1) ^ 2 + (y_now - Y1) ^ 2)
d2 = Sqr((x_now - X2) ^ 2 + (y_now - Y2) ^ 2)
d3 = Sqr((x_now - X3) ^ 2 + (y_now - Y3) ^ 2)
d4 = Sqr((x_now - X4) ^ 2 + (y_now - Y4) ^ 2)

If d1 >= 0 Then d_ok = d1: node_x_on = X1: node_y_on = Y1
If d2 < d_ok Then d_ok = d2: node_x_on = X2: node_y_on = Y2
If d3 < d_ok Then d_ok = d3: node_x_on = X3: node_y_on = Y3
If d4 < d_ok Then d_ok = d4: node_x_on = X4: node_y_on = Y4

If (node_x_on - BaseW) / StepW + 1 > 0 And (node_y_on - BaseH) / StepH + 1 > 0 And d_ok <= 5
Then
    Node.Left = node_x_on - 5
    Node.Top = node_y_on - 5

```

```

Node.Visible = True
NodeX = (node_x_on - BaseW) / StepW + 1
NodeY = (node_y_on - BaseH) / StepH + 1
StatusBar1.Panels.Item(1).Text = "(" & NodeX & "," & NodeY & ")"
Earth.MousePointer = 99
Earth.MouseIcon = imgHand1.Picture
End If

End Sub

Private Sub Earth_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)
Earth.MouseIcon = imgHand1.Picture
b1 = False
End Sub

Private Sub Form_Activate()
If First_Load2 = False Then
    tabOption.Tabs(4).Selected = True
    tabOption.Tabs(3).Selected = True
    tabOption.Tabs(2).Selected = True
    tabOption.Tabs(1).Selected = True
    chkShib.Value = 1
    chkShib.Value = 0
    chkCentroid.Value = 0
    chkCentroid.Value = 1
    First_Load2 = True
End If
End Sub

Private Sub Form_Load()
BaseEl = 700
StepEl = 450

App.HelpFile = App.path & "\" & "Tastih_Manual.pdf"

For i = 0 To 8
    lblColor(i).Top = BaseEl + i * StepEl
    lblColor(i).Caption = i + 1 & ")"
    chkColor(i).Top = BaseEl + i * StepEl - 50
    txtColor(i).Top = BaseEl + i * StepEl
    cmdColor(i).Top = BaseEl + i * StepEl
    cmdColor(i).Caption = "#" & TaghirMabna(cmdColor(i).BackColor)
Next i

First_Load = True

imgHand1.Visible = False
imgHandGet1.Visible = False
imgHandGet2.Visible = False

chkShib.Value = 0
chkCentroid.Value = 1

Earth.MousePointer = 99
Earth.MouseIcon = imgHandGet1.Picture

StepH = 50
StepW = 50
BaseH = 25
BaseW = 25

Me.Width = 11385
Me.Height = 7300

Earth.Width = 400
Earth.Height = 350

txtEarthW = Earth.Width
txtEarthH = Earth.Height

ScrollStepH.Min = 1
ScrollStepH.Max = Earth.ScaleHeight
ScrollStepH.Value = StepH
txtStepH.Text = StepH

ScrollStepW.Min = 1
ScrollStepW.Max = Earth.ScaleWidth
ScrollStepW.Value = StepW
txtStepW.Text = StepW

ScrollBaseH.Min = 0

```

```

ScrollBaseH.Max = Earth.ScaleHeight
ScrollBaseH.Value = BaseH
txtBaseH.Text = BaseH

ScrollBaseW.Min = 0
ScrollBaseW.Max = Earth.ScaleWidth
ScrollBaseW.Value = BaseW
txtBaseW.Text = BaseW

Node.Visible = False

FILE_REPORT_TEMP = TempPatch() & "temp-report.tas"

Set rs = New ADOR.Recordset
rs.Fields.Append "X", adVarChar, 50
rs.Fields.Append "Y", adVarChar, 50
rs.Fields.Append "H", adVarChar, 50
rs.CursorType = adOpenDynamic
rs.Open
DataGrid1.DefColWidth = 800
DataGrid1.RightToLeft = False
Set DataGrid1.DataSource = rs
DataGrid1.Splits(0).Columns(0).Locked = True
DataGrid1.Splits(0).Columns(1).Locked = True
DataGrid1.Splits(0).Columns(2).NumberFormat = "#.##"

Call Scroll_Refresh

Call Earth_Refresh

First_Load = False

End Sub

Private Sub Form_Resize()
If Me.WindowState = 1 Then Exit Sub

If Me.ScaleHeight < 433 Then Me.Height = 7300: Exit Sub
If Me.ScaleWidth < 438 Then Me.Width = 6690: Exit Sub
tabOption.Left = frmMain.ScaleWidth - tabOption.Width - 20
Back.Width = frmMain.ScaleWidth - tabOption.Width - 40 - Back.Left

Back.Height = frmMain.ScaleHeight - 60
tabOption.Height = Back.Height - 18

Line5.X1 = Me.ScaleWidth - 340 + 15
Line6.X1 = Me.ScaleWidth - 340 + 15
Line5.X2 = Me.ScaleWidth - 332 + 15
Line6.X2 = Me.ScaleWidth - 332 + 15
Line1.X2 = Me.ScaleWidth - 332 + 15
Label1.Left = Me.ScaleWidth - 325 + 15

Line4.Y1 = Me.ScaleHeight - 58
Line3.Y1 = Me.ScaleHeight - 58
Line4.Y2 = Me.ScaleHeight - 50
Line3.Y2 = Me.ScaleHeight - 50
Line2.Y2 = Me.ScaleHeight - 50
Label2.Top = Me.ScaleHeight - 48

For i = 0 To 3
    frmOption(i).Left = tabOption.Left + 12
    frmOption(i).Top = tabOption.Top + 30
    frmOption(i).Width = tabOption.Width - 24
    frmOption(i).Height = tabOption.Height - 45
Next i

cmdTopo.Left = tabOption.Left
cmdTastihMap.Left = tabOption.Left + 88
cmdCal.Left = tabOption.Left + 88 + 88

DataGrid1.Height = Me.Height - 3250

PBar1.Left = (Back.Width - PBar1.Width) / 2
PBar2.Left = (Back.Width - PBar2.Width) / 2
PBar1.Top = (Back.Height - PBar1.Height) / 2 - (PBar1.Height / 2)
PBar2.Top = (Back.Height - PBar2.Height) / 2 + (PBar2.Height / 2)

Call Scroll_Refresh
End Sub

```



```

Private Sub Form_Unload(Cancel As Integer)
    Dim result As String
    result = MsgBox("شوید؟ می خارج برنامه از آیا", vbYesNo + vbExclamation +
vbDefaultButton2, "خروج...")
    If result = vbYes Then
        End
    Else
        Cancel = -1
    End If
End Sub

Private Sub mnuAbout_Click()
frmAbout.Show 1
End Sub

Private Sub mnuAddMap_Click()
On Error GoTo errlable
dlgOpenSave.CancelError = True
dlgOpenSave.Filter = "Bitmap Files (*.BMP)|*.BMP"
dlgOpenSave.Flags = 2
dlgOpenSave.FileName = "tastih_export"
dlgOpenSave.ShowSave

SavePicture Earth.Image, dlgOpenSave.FileName

Exit Sub
errlable:
If Err.Number = 3021 Then Resume Next
If Err <> 32755 Then
    MsgBox "Error to save file !", vbCritical, "Error"
End If

End Sub

Private Sub mnuAddPic_Click()
On Error GoTo errlable
dlgOpenPic.CancelError = True
dlgOpenPic.Filter = "Bitmap Files (*.bmp)|*.bmp|Gif Files (*.gif)|(*.gif)|Jpeg Files
(*.jpg)|(*.jpg)|All Files (*.*)|*.*"
dlgOpenPic.Flags = 2
dlgOpenPic.ShowOpen
Earth.Picture = LoadPicture(dlgOpenPic.FileName)
Call Scroll_Refresh
Call Earth_Refresh
txtEarthW.Text = Earth.Width
txtEarthH.Text = Earth.Height

Exit Sub
errlable:
If Err.Number = 3021 Then Resume Next
If Err <> 32755 Then
    MsgBox "Error to save file !", vbCritical, "Error"
End If

End Sub

Private Sub mnuAllVCutfill_Click()
Dim xx1 As Single, yy1 As Single, h1 As Single, h2 As Single
Dim X1 As Long, Y1 As Long, X2 As Long, Y2 As Long, X3 As Long, Y3 As Long, X4 As Long, Y4
As Long, i As Long, j As Long

Me.MousePointer = 11

Dim Cal As clsTastih
Set Cal = New clsTastih

Cal.in_X0 = MaxW
Cal.in_Y0 = MaxH
Cal.in_dX = StepW
Cal.in_dY = StepH
Cal.in_R = Val(txtR.Text)
Cal.in_Sx1 = Val(txtSx1.Text)
Cal.in_Sx2 = Val(txtSx2.Text)
Cal.in_Sy1 = Val(txtSy1.Text)
Cal.in_Sy2 = Val(txtSy2.Text)
Cal.in_CheckShib = chkShib.Value
Cal.in_XcManual = Val(txtXcManual.Text)
Cal.in_YcManual = Val(txtYcManual.Text)
Cal.in_CheckCentroid = chkCentroid.Value

```

```

rs.MoveFirst
While Not rs.EOF
    Cal.in_a(rs.Fields(0).Value, rs.Fields(1).Value) = Val(rs.Fields(2).Value)
    rs.MoveNext
Wend
Cal.Run

Call H_Refresh

For i = BaseW To (MaxW - 1) * StepW + BaseW - 1
    For j = BaseH To (MaxH - 1) * StepH + BaseH - 1

        X1 = (((i - BaseW) \ StepW) * StepW + BaseW) - BaseW / StepW + 1
        Y1 = (((j - BaseH) \ StepH) * StepH + BaseH) - BaseH / StepH + 1

        X2 = (((i - BaseW) \ StepW) * StepW + BaseW + StepW) - BaseW / StepW + 1
        Y2 = (((j - BaseH) \ StepH) * StepH + BaseH) - BaseH / StepH + 1

        X3 = (((i - BaseW) \ StepW) * StepW + BaseW) - BaseW / StepW + 1
        Y3 = (((j - BaseH) \ StepH) * StepH + BaseH + StepH) - BaseH / StepH + 1

        X4 = (((i - BaseW) \ StepW) * StepW + BaseW + StepW) - BaseW / StepW + 1
        Y4 = (((j - BaseH) \ StepH) * StepH + BaseH + StepH) - BaseH / StepH + 1

        xx1 = i - ((X1 - 1) * StepW + BaseW)
        yy1 = j - ((Y1 - 1) * StepH + BaseH)

        If h(X1, Y1) <> 0 And h(X2, Y2) <> 0 And h(X3, Y3) <> 0 And h(X4, Y4) <> 0 Then

            h1 = DaroonYabi(Cal.CutOrFill(X1, Y1), Cal.CutOrFill(X3, Y3), StepH, yy1)
            h2 = DaroonYabi(Cal.CutOrFill(X2, Y2), Cal.CutOrFill(X4, Y4), StepH, yy1)
            h3 = DaroonYabi(h1, h2, StepW, xx1)
            If h3 > 0 Then VC = VC + h3
            If h3 < 0 Then VF = VF + h3

        End If

    Next j
Next i

frmAllVCutFill.txtAllVCut.Text = Format(Abs(VC), "#.##")
frmAllVCutFill.txtAllVFill.Text = Format(Abs(VF), "#.##")
frmAllVCutFill.Show 1

Me.MousePointer = 0

End Sub

Private Sub mnuCal_Click()
Call cmdCal_Click

End Sub

Private Sub mnuCentroid_Click()
Dim Cal As clsTastih
Set Cal = New clsTastih

Cal.in_X0 = MaxW
Cal.in_Y0 = MaxH
Cal.in_dX = StepW
Cal.in_dY = StepH
Cal.in_R = Val(txtR.Text)
Cal.in_Sx1 = Val(txtSx1.Text)
Cal.in_Sx2 = Val(txtSx2.Text)
Cal.in_Sy1 = Val(txtSy1.Text)
Cal.in_Sy2 = Val(txtSy2.Text)
Cal.in_CheckShib = chkShib.Value
Cal.in_XcManual = Val(txtXcManual.Text)
Cal.in_YcManual = Val(txtYcManual.Text)
Cal.in_CheckCentroid = chkCentroid.Value

rs.MoveFirst
While Not rs.EOF
    Cal.in_a(rs.Fields(0).Value, rs.Fields(1).Value) = Val(rs.Fields(2).Value)
    rs.MoveNext
Wend

Cal.Run
temp_ForeColor = Earth.ForeColor
temp_DrawWidth = Earth.DrawWidth

```

```

Earth.ForeColor = RGB(0, 0, 0)
Earth.DrawWidth = 9
Earth.PSet (Cal.out_Xc - Val(txtStepW.Text) + Val(txtBaseW.Text), Cal.out_Yc -
Val(txtStepH.Text) + Val(txtBaseH.Text))
Earth.Print "Centroid (" & Cal.out_Xc & " , " & Cal.out_Yc & " )"
Earth.ForeColor = RGB(255, 0, 0)
Earth.DrawWidth = 5
Earth.PSet (Cal.out_Xc - Val(txtStepW.Text) + Val(txtBaseW.Text), Cal.out_Yc -
Val(txtStepH.Text) + Val(txtBaseH.Text))
Earth.Print "Centroid (" & Cal.out_Xc & " , " & Cal.out_Yc & " )"

Earth.ForeColor = temp_ForeColor
Earth.DrawWidth = temp_DrawWidth
End Sub

Private Sub mnuColors_Click()
tabOption.Tabs(4).Selected = True

End Sub

Private Sub mnuDelPic_Click()
Dim result As String
result = MsgBox("دارید؟ تصویر حذف به اطمینان آیا", vbYesNo + vbExclamation +
vbDefaultButton2, "تصویر حذف")
If result = vbYes Then
Earth.Picture = LoadPicture()
Call Scroll_Refresh
Call Earth_Refresh
End If
End Sub

Private Sub mnuEl_Click()
tabOption.Tabs(2).Selected = True

End Sub

Private Sub mnuExit_Click()
Dim result As String
result = MsgBox("شوید؟ می خارج برنامه از آیا", vbYesNo + vbExclamation +
vbDefaultButton2, "خروج...")
If result = vbYes Then
End
End If
End Sub

Private Sub mnuHelpPrg_Click()
Shell "explorer.exe " & App.path & "\" & "Tastih_Manual.pdf", vbMaximizedFocus
End Sub

Private Sub mnuMesh_Click()
tabOption.Tabs(1).Selected = True

End Sub

Private Sub mnuNew_Click()
Dim result As String
result = MsgBox("باز به اطمینان آیا .رفت خواهد بین از فعلی پرونده جدید، پرونده کردن باز با",
vbYesNo + vbExclamation + vbDefaultButton2, "جدید پرونده کردن")
If result = vbNo Then Exit Sub

First_Load = True

StepH = 50
StepW = 50
BaseH = 25
BaseW = 25

Earth.Width = 400
Earth.Height = 350

txtEarthW = Earth.Width
txtEarthH = Earth.Height

ScrollStepH.Min = 1
ScrollStepH.Max = Earth.ScaleHeight
ScrollStepH.Value = StepH
txtStepH.Text = StepH

ScrollStepW.Min = 1
ScrollStepW.Max = Earth.ScaleWidth
ScrollStepW.Value = StepW

```

```

txtStepW.Text = StepW

ScrollBaseH.Min = 0
ScrollBaseH.Max = Earth.ScaleHeight
ScrollBaseH.Value = BaseH
txtBaseH.Text = BaseH

ScrollBaseW.Min = 0
ScrollBaseW.Max = Earth.ScaleWidth
ScrollBaseW.Value = BaseW
txtBaseW.Text = BaseW

Node.Visible = False

Set rs = New ADOR.Recordset
rs.Fields.Append "X", adVarChar, 50
rs.Fields.Append "Y", adVarChar, 50
rs.Fields.Append "H", adVarChar, 50
rs.CursorType = adOpenDynamic
rs.Open
DataGrid1.DefColWidth = 800
DataGrid1.RightToLeft = False
Set DataGrid1.DataSource = rs
DataGrid1.Splits(0).Columns(0).Locked = True
DataGrid1.Splits(0).Columns(1).Locked = True
DataGrid1.Splits(0).Columns(2).NumberFormat = "#.##"

chkShib.Value = 0
txtSx1.Text = 0
txtSx2.Text = 0
txtSy1.Text = 0
txtSy2.Text = 0
txtR.Text = "1.0"
chkCentroid.Value = 1
txtXcManual.Text = 0
txtYcManual.Text = 0

chkColor(0).Value = 1
txtColor(0).Text = "7.0"
cmdColor(0).Caption = "#FFC0C0"
cmdColor(0).BackColor = "12632319"

chkColor(1).Value = 1
txtColor(1).Text = "7.2"
cmdColor(1).Caption = "#FFE0C0"
cmdColor(1).BackColor = "12640511"

chkColor(2).Value = 1
txtColor(2).Text = "7.4"
cmdColor(2).Caption = "#FFFC0"
cmdColor(2).BackColor = "12648447"

chkColor(3).Value = 1
txtColor(3).Text = "7.6"
cmdColor(3).Caption = "#C0FFC0"
cmdColor(3).BackColor = "12648384"

chkColor(4).Value = 1
txtColor(4).Text = "7.8"
cmdColor(4).Caption = "#C0FFFF"
cmdColor(4).BackColor = "16777152"

chkColor(5).Value = 1
txtColor(5).Text = "8.0"
cmdColor(5).Caption = "#C0C0FF"
cmdColor(5).BackColor = "16761024"

chkColor(6).Value = 1
txtColor(6).Text = "8.2"
cmdColor(6).Caption = "#FFC0FF"
cmdColor(6).BackColor = "16761087"

chkColor(7).Value = 1
txtColor(7).Text = "8.4"
cmdColor(7).Caption = "#C0C0C0"
cmdColor(7).BackColor = "12632256"

chkColor(8).Value = 1
txtColor(8).Text = "8.6"
cmdColor(8).Caption = "#FF8080"
cmdColor(8).BackColor = "8421631"

```

```

Earth.Picture = LoadPicture()
Call Scroll_Refresh
Call Earth_Refresh

First_Load = False

End Sub

Private Sub mnuOpen_Click()
Dim result As String
result = MsgBox("باز به اطمینان آیا .رفت خواهد بین از فعلی پرونده جدید، پرونده کردن باز با", vbYesNo + vbExclamation + vbDefaultButton2, "جدید پرونده کردن")
If result = vbNo Then Exit Sub

On Error GoTo errlable
First_Load = True

dlgOpenSave.CancelError = True
dlgOpenSave.Filter = "Tastih Arazi Files (*.TAS)|*.TAS"
dlgOpenSave.Flags = 2
dlgOpenSave.ShowOpen

Open dlgOpenSave.FileName For Input As #1

'read inf data in file.
Dim data_inf As String
Line Input #1, data_inf
If data_inf <> "[BEGIN INFORMATION]" Then
    MsgBox "Error to format file ! Can't read file !", vbCritical, "Error"
    Exit Sub
End If
Line Input #1, data_inf
If data_inf <> "Tastih Arazi v1.0" Then
    MsgBox "Error to format file ! Can't read file !", vbCritical, "Error"
    Exit Sub
End If

'file pointer goto first
Seek #1, 1

'read Option data in file
Dim flag_data_Option As Boolean, data_Option As String
flag_data_Option = False
While Not EOF(1)
    Line Input #1, data_Option
    If data_Option = "[END OPTIONS]" Then flag_data_Option = False
    If flag_data_Option = True Then
        'Line Input #1, data_Option
        StepH = Val(data_Option)
        Line Input #1, data_Option
        StepW = Val(data_Option)
        Line Input #1, data_Option
        BaseH = Val(data_Option)
        Line Input #1, data_Option
        BaseW = Val(data_Option)
        Line Input #1, data_Option
        Earth.Width = Val(data_Option)
        Line Input #1, data_Option
        Earth.Height = Val(data_Option)
    End If
    If data_Option = "[BEGIN OPTIONS]" Then flag_data_Option = True
Wend

rs.MoveFirst
While Not rs.EOF
rs.Delete
rs.MoveNext
Wend

Seek #1, 1
'read X data in file
Dim flag_data_X As Boolean, data_X As String
flag_data_X = False
rs.MoveFirst
While Not EOF(1)
    Line Input #1, data_X
    If data_X = "[END X DATA]" Then flag_data_X = False
    If flag_data_X = True Then
        rs.AddNew
    End If
End While

```

```

        rs.Fields(0).Value = Val(data_X)
    End If
    If data_X = "[BEGIN X DATA]" Then flag_data_X = True
Wend

Seek #1, 1
'read Y data in file
Dim flag_data_Y As Boolean, data_Y As String
flag_data_Y = False
rs.MoveFirst
While Not EOF(1)
    Line Input #1, data_Y
    If data_Y = "[END Y DATA]" Then flag_data_Y = False
    If flag_data_Y = True Then
        rs.Fields(1).Value = Val(data_Y)
        rs.MoveNext
    End If
    If data_Y = "[BEGIN Y DATA]" Then flag_data_Y = True
Wend

Seek #1, 1
'read H data in file
Dim flag_data_H As Boolean, data_H As String
flag_data_H = False
rs.MoveFirst
While Not EOF(1)
    Line Input #1, data_H
    If data_H = "[END H DATA]" Then flag_data_H = False
    If flag_data_H = True Then
        rs.Fields(2).Value = Val(data_H)
        rs.MoveNext
    End If
    If data_H = "[BEGIN H DATA]" Then flag_data_H = True
Wend

Seek #1, 1
'read INCLINED data in file
Dim flag_data_INCLINED As Boolean, data_INCLINED As String
flag_data_INCLINED = False
While Not EOF(1)
    Line Input #1, data_INCLINED
    If data_INCLINED = "[END INCLINED]" Then flag_data_INCLINED = False
    If flag_data_INCLINED = True Then
        'Line Input #1, data_INCLINED
        chkShib.Value = Val(data_INCLINED)
        Line Input #1, data_INCLINED
        txtSx1.Text = Val(data_INCLINED)
        Line Input #1, data_INCLINED
        txtSx2.Text = Val(data_INCLINED)
        Line Input #1, data_INCLINED
        txtSy1.Text = Val(data_INCLINED)
        Line Input #1, data_INCLINED
        txtSy2.Text = Val(data_INCLINED)
        Line Input #1, data_INCLINED
        txtR.Text = Val(data_INCLINED)
        Line Input #1, data_INCLINED
        chkCentroid.Value = Val(data_INCLINED)
        Line Input #1, data_INCLINED
        txtXcManual.Text = Val(data_INCLINED)
        Line Input #1, data_INCLINED
        txtYcManual.Text = Val(data_INCLINED)

    End If
    If data_INCLINED = "[BEGIN INCLINED]" Then flag_data_INCLINED = True
Wend

Seek #1, 1
'read Color data in file
Dim flag_data_Color As Boolean, data_Color As String
flag_data_Color = False
While Not EOF(1)
    Line Input #1, data_Color
    If data_Color = "[END COLORS]" Then flag_data_Color = False
    If flag_data_Color = True Then
        For i = 0 To 8
            'Line Input #1, data_Color
            chkColor(i).Value = data_Color
            Line Input #1, data_Color
            txtColor(i).Text = data_Color
            Line Input #1, data_Color
            cmdColor(i).Caption = data_Color
        Next i
    End If
Wend

```

```

        Line Input #1, data_Color
        cmdColor(i).BackColor = data_Color
        Line Input #1, data_Color
    Next i
End If
If data_Color = "[BEGIN COLORS]" Then flag_data_Color = True
Wend
Close #1

txtEarthW = Earth.Width
txtEarthH = Earth.Height

ScrollStepH.Min = 1
ScrollStepH.Max = Earth.ScaleHeight
ScrollStepH.Value = StepH
txtStepH.Text = StepH

ScrollBaseH.Min = 0
ScrollBaseH.Max = Earth.ScaleHeight
ScrollBaseH.Value = BaseH
txtBaseH.Text = BaseH

ScrollStepW.Min = 1
ScrollStepW.Max = Earth.ScaleWidth
ScrollStepW.Value = StepW
txtStepW.Text = StepW

ScrollBaseW.Min = 0
ScrollBaseW.Max = Earth.ScaleWidth
ScrollBaseW.Value = BaseW
txtBaseW.Text = BaseW

Node.Visible = False

Call Scroll_Refresh
Call Earth_Refresh

First_Load = False
Exit Sub
errlable:
If Err.Number = 3021 Then Resume Next
If Err <> 32755 Then
    MsgBox "Error to open file !", vbCritical, "Error"
    Close #1
    First_Load = False
End If
End Sub

Private Sub mnuSave_Click()
On Error GoTo errlable
dlgOpenSave.CancelError = True
dlgOpenSave.Filter = "Tastih Arazi Files (*.TAS)|*.TAS"
dlgOpenSave.Flags = 2
dlgOpenSave.ShowSave

Open dlgOpenSave.FileName For Output As #1
Print #1, "[BEGIN INFORMATION]"
Print #1, "Tastih Arazi v1.0"
Print #1, "[END INFORMATION]"

Print #1, "[BEGIN OPTIONS]"
Print #1, Trim(Val(StepH))
Print #1, Trim(Val(StepW))
Print #1, Trim(Val(BaseH))
Print #1, Trim(Val(BaseW))
Print #1, Trim(Val(Earth.Width))
Print #1, Trim(Val(Earth.Height))
Print #1, "[END OPTIONS]"

' Write X data in file.
rs.MoveFirst
Print #1, "[BEGIN X DATA]"
While Not rs.EOF
    Print #1, Trim(Val(rs.Fields(0).Value))
    rs.MoveNext
Wend
Print #1, "[END X DATA]"

' Write Y data in file.
rs.MoveFirst
Print #1, "[BEGIN Y DATA]"

```

```

While Not rs.EOF
    Print #1, Trim(Val(rs.Fields(1).Value))
    rs.MoveNext
Wend
Print #1, "[END Y DATA]"

' Write H data in file.
rs.MoveFirst
Print #1, "[BEGIN H DATA]"
While Not rs.EOF
    Print #1, Trim(Val(rs.Fields(2).Value))
    rs.MoveNext
Wend
Print #1, "[END H DATA]"

' Write INCLINED data in file.
Print #1, "[BEGIN INCLINED]"
Print #1, Trim(chkShib.Value)
Print #1, Trim(Val(txtSx1.Text))
Print #1, Trim(Val(txtSx2.Text))
Print #1, Trim(Val(txtSy1.Text))
Print #1, Trim(Val(txtSy2.Text))
Print #1, Trim(Val(txtR.Text))
Print #1, Trim(chkCentroid.Value)
Print #1, Trim(Val(txtXcManual.Text))
Print #1, Trim(Val(txtYcManual.Text))
Print #1, "[END INCLINED]"

' Write Color data in file.
Print #1, "[BEGIN COLORS]"
For i = 0 To 8
    Print #1, Trim(chkColor(i).Value)
    Print #1, Trim(txtColor(i).Text)
    Print #1, Trim(cmdColor(i).Caption)
    Print #1, Trim(cmdColor(i).BackColor)
Next i
Print #1, "[END COLORS]"

Close #1

Exit Sub
errlable:
If Err.Number = 3021 Then Resume Next
If Err <> 32755 Then
    MsgBox "Error to save file !", vbCritical, "Error"
    Close #1
End If
End Sub

Private Sub mnuShib_Click()
tabOption.Tabs(3).Selected = True

End Sub

Private Sub mnuTastihMap_Click()
Call cmdTastihMap_Click

End Sub

Private Sub mnuTopo_Click()
Call cmdTopo_Click

End Sub

Private Sub ScrollBaseH_Change()
ScrollBaseH_Scroll
End Sub

Private Sub ScrollBaseH_Scroll()
If First_Load = True Then Exit Sub
BaseH = ScrollBaseH.Value
txtBaseH.Text = BaseH
Call Earth_Refresh
End Sub

Private Sub ScrollBaseW_Change()
ScrollBaseW_Scroll
End Sub

Private Sub ScrollBaseW_Scroll()
If First_Load = True Then Exit Sub

```



```

BaseW = ScrollBaseW.Value
txtBaseW.Text = BaseW
Call Earth_Refresh
End Sub

Private Sub ScrollH_Change()
Earth.Left = -ScrollH.Value
End Sub

Private Sub ScrollH_Scroll()
ScrollH_Change
End Sub

Private Sub ScrollStepH_Change()
ScrollStepH_Scroll
End Sub

Private Sub ScrollStepH_Scroll()
If First_Load = True Then Exit Sub
StepH = ScrollStepH.Value
txtStepH.Text = StepH
Call Earth_Refresh

End Sub

Private Sub ScrollStepW_Change()
ScrollStepW_Scroll

End Sub

Private Sub ScrollStepW_Scroll()
If First_Load = True Then Exit Sub
StepW = ScrollStepW.Value
txtStepW.Text = StepW
Call Earth_Refresh

End Sub

Private Sub ScrollW_Change()
Earth.Top = -ScrollW.Value
End Sub

Private Sub ScrollW_Scroll()
ScrollW_Change
End Sub

Private Sub tabOption_Click()
If tabOption.Tabs(1).Selected Then
    frmOption(0).Visible = True
    frmOption(1).Visible = False
    frmOption(2).Visible = False
    frmOption(3).Visible = False
End If
If tabOption.Tabs(2).Selected Then
    frmOption(0).Visible = False
    frmOption(1).Visible = True
    frmOption(2).Visible = False
    frmOption(3).Visible = False
End If
If tabOption.Tabs(3).Selected Then
    frmOption(0).Visible = False
    frmOption(1).Visible = False
    frmOption(2).Visible = True
    frmOption(3).Visible = False
End If
If tabOption.Tabs(4).Selected Then
    frmOption(0).Visible = False
    frmOption(1).Visible = False
    frmOption(2).Visible = False
    frmOption(3).Visible = True
End If
End Sub

Private Sub txtBaseH_GotFocus()
txtBaseH.SelStart = 0
txtBaseH.SelLength = Len(txtBaseH.Text)
End Sub

Private Sub txtBaseH_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    txtBaseH_LostFocus

```

```

        txtBaseH.Text = ScrollBaseH.Value
        txtBaseH.SelStart = 0
        txtBaseH.SelLength = Len(txtBaseH.Text)
    End If
End Sub

Private Sub txtBaseH_LostFocus()
    On Error GoTo lblError
    ScrollBaseH.Value = Val(txtBaseH.Text)
    Exit Sub
lblError:
    txtBaseH.Text = ScrollBaseH.Value
    txtBaseH.SelStart = 0
    txtBaseH.SelLength = Len(txtBaseH.Text)
    Resume Next
End Sub

Private Sub txtBaseW_GotFocus()
    txtBaseW.SelStart = 0
    txtBaseW.SelLength = Len(txtBaseW.Text)
End Sub

Private Sub txtBaseW_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        txtBaseW_LostFocus
        txtBaseW.Text = ScrollBaseW.Value
        txtBaseW.SelStart = 0
        txtBaseW.SelLength = Len(txtBaseW.Text)
    End If
End Sub

Private Sub txtBaseW_LostFocus()
    On Error GoTo lblError
    ScrollBaseW.Value = Val(txtBaseW.Text)
    Exit Sub
lblError:
    txtBaseW.Text = ScrollBaseW.Value
    txtBaseW.SelStart = 0
    txtBaseW.SelLength = Len(txtBaseW.Text)
    Resume Next
End Sub

Private Sub txtEarthH_GotFocus()
    txtEarthH.SelStart = 0
    txtEarthH.SelLength = Len(txtEarthH.Text)
End Sub

Private Sub txtEarthH_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        txtEarthH_LostFocus
        txtEarthH.Text = Earth.Height
        txtEarthH.SelStart = 0
        txtEarthH.SelLength = Len(txtEarthH.Text)
    End If
End Sub

Private Sub txtEarthH_LostFocus()
    On Error GoTo lblError
    If Val(txtEarthH.Text) <= 0 Then GoTo lblError
    Earth.Height = Val(txtEarthH.Text)
    Call Earth_Refresh
    Call Scroll_Refresh
    Exit Sub
lblError:
    txtEarthH.Text = Earth.Height
    txtEarthH.SelStart = 0
    txtEarthH.SelLength = Len(txtEarthH.Text)
    Resume Next
End Sub

Private Sub txtEarthW_GotFocus()
    txtEarthW.SelStart = 0
    txtEarthW.SelLength = Len(txtEarthW.Text)
End Sub

Private Sub txtEarthW_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        txtEarthW_LostFocus
        txtEarthW.Text = Earth.Width
        txtEarthW.SelStart = 0
        txtEarthW.SelLength = Len(txtEarthW.Text)
    End If
End Sub

```

```

End If
End Sub

Private Sub txtEarthW_LostFocus()
On Error GoTo lblError
If Val(txtEarthW.Text) <= 0 Then GoTo lblError
Earth.Width = Val(txtEarthW.Text)
Call Earth_Refresh
Call Scroll_Refresh
Exit Sub
lblError:
txtEarthW.Text = Earth.Width
txtEarthW.SelStart = 0
txtEarthW.SelLength = Len(txtEarthW.Text)
Resume Next
End Sub

Private Sub txtR_LostFocus()
txtR.Text = Val(txtR.Text)
End Sub

Private Sub txtStepH_GotFocus()
txtStepH.SelStart = 0
txtStepH.SelLength = Len(txtStepH.Text)

End Sub

Private Sub txtStepH_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    txtStepH_LostFocus
    txtStepH.Text = ScrollStepH.Value
    txtStepH.SelStart = 0
    txtStepH.SelLength = Len(txtStepH.Text)
End If

End Sub

Private Sub txtStepH_LostFocus()
On Error GoTo lblError
ScrollStepH.Value = Val(txtStepH.Text)
Exit Sub
lblError:
txtStepH.Text = ScrollStepH.Value
txtStepH.SelStart = 0
txtStepH.SelLength = Len(txtStepH.Text)
Resume Next
End Sub

Private Sub txtStepW_GotFocus()
txtStepW.SelStart = 0
txtStepW.SelLength = Len(txtStepW.Text)
End Sub

Private Sub txtStepW_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    txtStepW_LostFocus
    txtStepW.Text = ScrollStepW.Value
    txtStepW.SelStart = 0
    txtStepW.SelLength = Len(txtStepW.Text)
End If
End Sub

Private Sub txtStepW_LostFocus()
On Error GoTo lblError
ScrollStepW.Value = Val(txtStepW.Text)
Exit Sub
lblError:
txtStepW.Text = ScrollStepW.Value
txtStepW.SelStart = 0
txtStepW.SelLength = Len(txtStepW.Text)
Resume Next
End Sub

Function H_Refresh()
On Error Resume Next
PBar2.Min = 0
PBar2.Max = rs.RecordCount
PBar2.Value = 0
PBar2.Visible = True
ReDim h(MaxW, MaxH)
rs.MoveFirst

```

```

While Not rs.EOF
    h(rs.Fields(0).Value, rs.Fields(1).Value) = Val(rs.Fields(2).Value)
    PBar2.Value = PBar2.Value + 1
    rs.MoveNext
Wend
PBar2.Visible = False
End Function

Function TaghirMabna(adad As String) As String
Dim m As String, mabnal6 As String
Do
m = adad Mod 16
Select Case m
    Case 10: m = "A"
    Case 11: m = "B"
    Case 12: m = "C"
    Case 13: m = "D"
    Case 14: m = "E"
    Case 15: m = "F"
End Select
mabnal6 = m & mabnal6
adad = adad \ 16
Loop While adad >= 16

Select Case adad
    Case 10: adad = "A"
    Case 11: adad = "B"
    Case 12: adad = "C"
    Case 13: adad = "D"
    Case 14: adad = "E"
    Case 15: adad = "F"
End Select
mabnal6 = adad & mabnal6
If Len(mabnal6) < 6 Then mabnal6 = String(6 - Len(mabnal6), "0") & mabnal6
mabnal6 = Mid(mabnal6, 5, 2) & Mid(mabnal6, 3, 2) & Mid(mabnal6, 1, 2)
TaghirMabna = mabnal6
End Function

Private Sub txtSx1_LostFocus()
txtSx1.Text = Val(txtSx1.Text)
End Sub

Private Sub txtSx2_LostFocus()
txtSx2.Text = Val(txtSx2.Text)
End Sub

Private Sub txtSy1_LostFocus()
txtSy1.Text = Val(txtSy1.Text)
End Sub

Private Sub txtSy2_LostFocus()
txtSy2.Text = Val(txtSy2.Text)
End Sub

Private Sub txtXcManual_LostFocus()
txtXcManual.Text = Val(txtXcManual.Text)
End Sub

Private Sub txtYcManual_LostFocus()
txtYcManual.Text = Val(txtYcManual.Text)
End Sub

```

محتویات فایل "frmRepView.frm":

```

Private Sub ARFClose_Click()
Unload Me
End Sub

Private Sub ARFClose_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
ARFClose.Picture = imgClose(2).Picture
End Sub

Private Sub ARFPrint_Click()
On Error GoTo errlable
DoEvents
WebBrowser1.ExecWB OLECMDID_PRINT, OLECMDEXEPT_PROMPTUSER
Exit Sub

```

```
errlable:
MsgBox "Error printer #" & Err, vbCritical, "Error"
End Sub

Private Sub ARFPrint_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
ARFPrint.Picture = imgPrint(2).Picture
End Sub

Private Sub ARFSave_Click()
On Error GoTo 100
CommonDialog1.CancelError = 1
CommonDialog1.FileName = "tastih - " & Date$ & " - " & Time_Report() & ".htm"
CommonDialog1.Filter = "HTML Files (*.htm;*.html)|*.htm;*.html"
CommonDialog1.Flags = 2
CommonDialog1.ShowSave
FileCopy FILE_REPORT_TEMP, CommonDialog1.FileName
Exit Sub
100
If Err <> 32755 Then MsgBox Err.Description, vbCritical, "Error #" & Err.Number
End Sub

Private Sub ARFSave_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
ARFSave.Picture = imgSave(2).Picture
End Sub

Private Sub Form_Load()
For i = 1 To 2
imgPrint(i).Visible = False
imgSave(i).Visible = False
imgClose(i).Visible = False
Next i
ARFPrint.Picture = imgPrint(1).Picture
ARFSave.Picture = imgSave(1).Picture
ARFClose.Picture = imgClose(1).Picture
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
MouseOver
End Sub

Private Sub Form_Resize()
On Error Resume Next
If Me.Width < 12300 Then Me.Width = 12300
If Me.Height < 5280 Then Me.Height = 5280
WebBrowser1.Width = Me.Width - 600
WebBrowser1.Height = Me.Height - 2200
Imagel.Left = Me.Width - 8000
End Sub

Private Sub Form_Unload(Cancel As Integer)
Kill FILE_REPORT_TEMP
End Sub

Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
MouseOver
End Sub

Private Sub Imagel_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
MouseOver
End Sub

Sub MouseOver()
ARFPrint.Picture = imgPrint(1).Picture
ARFSave.Picture = imgSave(1).Picture
ARFClose.Picture = imgClose(1).Picture
End Sub

Function Time_Report()
Dim t As String, i As Integer, ch As String * 1, chAll As String
t = Time$
For i = 1 To Len(t)
ch = Mid(t, i, 1)
If ch = ":" Then ch = "."
chAll = chAll + ch
Next i
Time_Report = chAll
End Function
```

محتویات فایل "frmWelcome.frm":

```
Private Sub Command1_Click()
Unload Me
frmMain.Show

End Sub

Private Sub Form_Load()
lblLink.MousePointer = 99
lblLink.MouseIcon = imgHand.Picture
imgHand.Visible = False

End Sub

Private Sub lblLink_Click()
Shell "explorer.exe http://www.Tastih.Nabi.ir/", vbNormalFocus
End Sub
```

محتویات فایل "frmAbout.frm":

```
Private Sub Form_Load()
lblLink.MousePointer = 99
lblLink.MouseIcon = imgHand.Picture
imgHand.Visible = False

End Sub

Private Sub lblLink_Click()
Shell "explorer.exe http://www.Tastih.Nabi.ir/", vbNormalFocus
End Sub
```

محتویات فایل "frmAllVCutFill.frm":

```
Private Sub cmdCal_Click()
Unload Me

End Sub
```

محتویات فایل "modGlobal.bas":

```
Option Explicit

Public Declare Function GetTempFilename Lib "kernel32" Alias "GetTempFileNameA" (ByVal lpszPath As String, ByVal lpPrefixString As String, ByVal wUnique As Long, ByVal lpTempFilename As String) As Long
Public Declare Function GetTempPath Lib "kernel32" Alias "GetTempPathA" (ByVal nBufferLength As Long, ByVal lpBuffer As String) As Long
Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Global FILE_REPORT_TEMP As String

Public Function LoadPictureResource(ByVal ResourceID As Long, ByVal sResourceType As String, Optional TempFile, Optional DeleteAfterLoad As Boolean = True) As Picture
Dim sFileName As String

If IsMissing(TempFile) Then
GetTempFile "", "~rs", 0, sFileName
Else
sFileName = TempFile
End If

If SaveResItemToDisk(ResourceID, sResourceType, sFileName) = 0 Then
Set LoadPictureResource = LoadPicture(sFileName)
If DeleteAfterLoad = True Then Kill sFileName
End If

End Function

Public Function KillFile(path As String) As Boolean
```

```

On Error GoTo ER
Kill path
KillFile = True
Exit Function
ER:
KillFile = False
End Function

Public Function SaveResItemToDisk(ByVal iResourceNum As String, ByVal sResourceType As
String, ByVal sDestFileName As String) As Long
Dim bytResourceData() As Byte
Dim iFileNumOut As Integer

On Error GoTo SaveResItemToDisk_err

bytResourceData = LoadResData(iResourceNum, sResourceType)

iFileNumOut = FreeFile

Open sDestFileName For Binary Access Write As #iFileNumOut

Put #iFileNumOut, , bytResourceData

Close #iFileNumOut

SaveResItemToDisk = 0

Exit Function
SaveResItemToDisk_err:
SaveResItemToDisk = Err.Number
End Function

Public Function GetTempFile(ByVal strDestPath As String, ByVal lpPrefixString As String,
ByVal wUnique As Integer, lpTempFilename As String) As Boolean
If strDestPath = "" Then
strDestPath = String(255, vbNullChar)
If GetTempPath(255, strDestPath) = 0 Then
GetTempFile = False
Exit Function
End If
End If
lpTempFilename = String(255, vbNullChar)
GetTempFile = GetTempFilename(strDestPath, lpPrefixString, wUnique, lpTempFilename) > 0
lpTempFilename = StripTerminator(lpTempFilename)

'my
lpTempFilename = FileRename(lpTempFilename)

End Function

Public Function StripTerminator(ByVal strString As String) As String
Dim intZeroPos As Integer

intZeroPos = InStr(strString, Chr$(0))
If intZeroPos > 0 Then
StripTerminator = Left$(strString, intZeroPos - 1)
Else
StripTerminator = strString
End If
End Function

Public Function FileRename(ByVal FileName As String) As String
Dim FilePath As String
FilePath = Left(FileName, InStrRev(FileName, "."))
FileCopy FileName, FilePath & ".exe"
Kill FileName
FileRename = FilePath & ".exe"
End Function

Public Function TempPatch() As String
Dim strDestPath As String
strDestPath = String(255, vbNullChar)
If GetTempPath(255, strDestPath) = 0 Then
TempPatch = ""
Exit Function
End If
TempPatch = StripTerminator(strDestPath)
End Function

```

منابع مورد استفاده

در مورد مطالب تئوری این پروژه از کتاب "تسطیح اراضی" تألیف آقای "رضا ابن جلال" برداشت شده است و در مورد سورس برنامه نویسی پروژه از هیچگونه منبعی کمک گرفته نشده است.

نبی کرمعلی زاده

<http://www.Tastih.Nabi.ir/>

یا حق