

نکات تکمیلی پروژه

۱. کتابخانه های مورد نیاز:

```
1 #include <stdio.h>
2 #include <windows.h>
3 #include <conio.h>
4 #include <time.h>
5 #include <ctime>
```

۲. تابعی جهت جابجا کردن مکان نما:

جهت چاپ یک کاراکتر در مکانی مشخص، قبل از فراخوانی printf یا سایر توابع باید فراخوانی شود.

```
1 void gotoxy(int x, int y) {
2     HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
3     COORD cursorCoord;
4     cursorCoord.X = y;
5     cursorCoord.Y = x;
6     SetConsoleCursorPosition(consoleHandle, cursorCoord);
7 }
```

۳. تابعی جهت پاک کردن صفحه نمایش:

```
1 void clearScreen() {
2     system("cls");
3 }
```

۴. تابعی جهت تغییر رنگ متن:

با فراخوانی این تابع قبل از چاپ متن مورد نظر و دادن کد رنگ مورد نظر از ۰ تا ۱۵ به عنوان پارامتر، رنگ نوشته به دلخواه تغییر خواهد کرد.

```
1 void setTextColor(int textColor, int backColor = 0) {
2     HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
3     int colorAttribute = backColor << 4 | textColor;
4     SetConsoleTextAttribute(consoleHandle, colorAttribute);
5 }
```

۵. تابعی جهت ایجاد delay:

با فراخوانی این تابع و پاس دادن عددی در واحد میلی ثانیه، کار برنامه به اندازه دلخواه متوقف میشود.

```
1 void sleep(unsigned int mseconds) {
2     clock_t goal = mseconds + clock();
3     while (goal > clock());
4 }
```

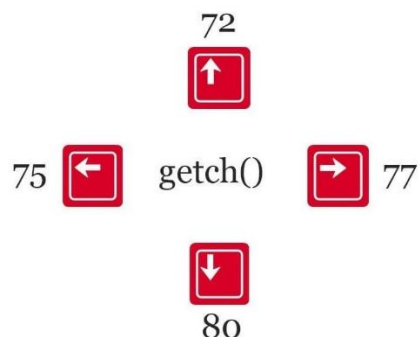
۶. برای بررسی فشردن شدن کلیدی از روی صفحه کلید میتوان از تابع `kbhit()` استفاده کرد. این تابع در صورتی که کلیدی روی صفحه کلید فشرده نشده باشد عدد صفر و در صورت فشرده شدن کلیدی روی صفحه کلید عددی غیر صفر باز می گرداند. این تابع در کتابخانه `conio.h` پیاده سازی شده است.

۷. برای محاسبه زمان میتوان از تابع `clock()` در کتابخانه `ctime` استفاده کرد. این تابع زمان سپری شده از شروع برنامه را بر حسب میلی ثانیه به شما باز میگرداند.

۸. برای به دست آوردن اعداد تصادفی میتوان از تابع `rand()` استفاده کرد. این تابع با استفاد از الگوریتمی هر بار عددی بین صفر تا `MAX_RANDOM` برمیگرداند. این الگوریتم وابسته به زمان اجرای تابع نیست در نتیجه شما اگر دو بار برنامه را اجرا کنید اعداد تصادفی مشابه را دریافت می کنید. برای رفع این محدودیت در ابتدای برنامه تنها یکبار تابع `srand(time(NULL))` را فراخوانی کنید. از این پس در هر بار فراخوانی تابع `rand()` اعداد متفاوتی را دریافت می کنید.

۹. کلیدهای جهت نما:

کلیدهای جهت نما (چپ، راست بالا و پایین) دو کده هستند. به این معنا که با فشردن این کلیدها دو کد پشت سر هم ارسال می شوند که کد اول بین آنها مشترک است! به عنوان مثال، با فشردن کلید جهت راست، فراخوانی اولین `getch()` دو مقدار ۲۲۴ و ۷۷ را باز می گرداند. و برای کلید جهت چپ مقادیر بازگشتی به ترتیب ۲۲۴ و ۷۵ هستند. بنابراین با این فرض که مقدار خروجی تابع `getch()` در متغیری از نوع `int` ذخیره شود؛ کافی است با یک شرط بررسی شود که اگر مقدار این متغیر برابر ۲۲۴ است دوباره تابع `getch()` فراخوانی شود. در این صورت تنها کد دوم را در متغیر مذکور خواهیم داشت.



۱۰. گیت:

- بر اساس الگوی git flow عمل کنید. (از [اینجا](#) مطالعه کنید)
- در نظر داشته باشید برنج main/master همواره شامل کد پایدار است. به طوری که هر زمان که clone شود، بتوان آن را بدون مشکل build و اجرا کرد.
- پیام commit ها را مناسب و با معنی انتخاب کنید. (از [اینجا](#) مطالعه کنید)
- فایل های اضافه همانند فایل های با پسوند exe، o، obj و ... را در فایل gitignore. لحاظ کنید تا در commit ها در نظر گرفته نشوند.

۱۱. کد تمیز:

- قواعد و ویژگی های کد تمیز را از [اینجا](#) مطالعه کنید.
- پروژه حتما باید متشکل از فایل های مختلف باشید.
- از توابع متعدد استفاده کنید و تا حد امکان آن ها را کوتاه و خوانا پیاده سازی کنید.
- هر تابع باید تنها یک کار مشخص را انجام دهد. از واگذاری وظایف و کارهای متعدد به یک تابع خودداری کنید.
- جهت افزایش خوانایی، از کامنت گذاری استفاده کنید.