

به نام خدا

دستور کار کارگاه کامپیوتر شماره ۵

توسعه یک برنامه نسبتاً بزرگ به صورت پیمانه‌ای

در این دستور کار قرار است یک ماشین حساب مهندسی را به صورت پیمانه‌ای پیاده سازی کنید. در ادامه بخش‌های مختلف این برنامه توضیح داده خواهد شد. قبل از این که پیاده‌سازی را شروع کنید دستور کار را کامل مطالعه کنید. پیاده‌سازی شما باید مبتنی بر دستور کار باشد در غیر این صورت نمره‌ای کسب نخواهید کرد!

توجه: برای پیاده‌سازی هیچ کدام از عملگرهای ریاضی، اجازه ندارید از کتابخانه‌های آماده مثل Math.h استفاده کنید!! برای محاسبات از فایل TaylorPolySeries.pdf استفاده کنید.

این ماشین حساب شامل عملگرهای زیر است:

- چهار عمل اصلی: ضرب (m)، تقسیم (d)، جمع (a) و تفریق (s).
- باقی‌مانده (M):  $x \% y$
- سینوس (S)
- کسینوس (C)
- تانژانت (t)
- توان‌رسانی (p):  $x^y$
- فاکتوریل (f):  $x!$
- معکوس عدد (r):  $\frac{1}{x}$
- تبدیل مبنای عدد (b):  $(a)_{b1} = (a')_{b2}$ ,  $(18)_{10} = (10010)_2$
- $\ln(x)$  (L): توجه کنید که محاسبات شما باید برای ورودی‌های خارج از بازه ذکر شده هم جواب داشته باشد.  $\log_e 5 = -\log_e \frac{1}{5}$
- امتیازی:
- ترکیب (c):  $\binom{n}{k}$
- جذر (q):  $\sqrt[y]{x}$

با توجه به توضیحات بالا مراحل زیر را باید پیاده‌سازی کنید:

۱. ورودی برنامه به این شکل است که کاربر یک حرف که نشان‌دهنده نوع تابع ریاضی است را به عنوان ورودی به برنامه وارد می‌کند. سپس با توجه به نوع عملگر که ممکن است نیازمند یک و یا دو عملوند باشد تعداد مورد نیاز عملوند را به عنوان ورودی‌های تابع وارد می‌کند. مثلاً برای محاسبه فاکتوریل عدد ۱۰ در ابتدا کاربر حرف f را وارد کرده و سپس عدد 10 را وارد می‌کند و خروجی برنامه می‌بایست مقدار 10! را محاسبه کند. ضمناً برنامه شما باید همواره در حالت آماده برای دریافت ورودی از کاربر باشد.

۲. هر کدام از عملگرهای ریاضی باید به صورت تابعی مجزا نوشته شوند. مثلاً یک تابع برای محاسبه سینوس و یا یک تابع برای محاسبه فاکتوریل نوشته شود. در صورت نوشتن اعمال درون تابع اصلی هیچ نمره‌ای به شما تعلق نخواهد گرفت.

۳. برای نمایش خروجی برنامه یک تابع بنویسید که با توجه به عملیات انجام شده خروجی مناسب چاپ کند. مثلاً اگر ورودی برنامه 5 f است خروجی برنامه به شکل 5 is 120 factorial of 5 باید باشد. تابع خروجی به شکل زیر است:

```
Void doPrint(char func, float in1, float in2, float result){
    if(func=='f'){
        printf("factorial of %f is %f", in1, result);
    }else if (...
        //Add your code here!
    }
```

نکته: برنامه می‌بایست تا زمانی که کاراکتر \* وارد نشده است، آماده دریافت ورودی از کاربر برای انجام محاسبات باشد. زمانی که کاربر ورودی \* را وارد می‌کند برنامه می‌بایست به کار خود پایان داده و خارج شود.

**راهنمایی:** برای پیاده‌سازی برنامه‌هایی از این دست که ویژگی‌های زیادی دارد به نکات زیر توجه کنید:

۱. از آنجایی که هر عملیات لازم است توسط یک تابع پیاده سازی شود. برای هر عملیات یک تابع تعریف کنید. این توابع در ابتدا خالی هستند و فقط ورودی و خروجی آنها مشخص شده است.

```
int factorial(int n){
    //You should add some code here and then change the "return n"!
    return n;
}
```

۲. تابع اصلی برنامه را بنویسید. این تابع ورودی‌ها را از کاربر خوانده و بر اساس اطلاعات، یکی از این توابع را فراخوانی می‌کند.

۳. برنامه نوشته شده را کامپایل و مشکلات syntax را رفع کنید .

۴. یکی از عملیات‌ها (قاعدتا ساده‌ترین آن‌ها) را انتخاب کرده و تابع مربوطه را تکمیل کنید. اگر این تابع پیچیده است آن را به چندین تابع کوچکتر بشکنید.

**نکته:** اگر برای پیاده سازی این عملیات، محاسباتی لازم است که در عملیات‌های دیگر هم لازم خواهد شد حتماً آن را به صورت تابع جداگانه پیاده سازی کنید.

۵. برنامه را کامپایل و اجرا نمایید، عملیاتی که پیاده سازی شده است را با چندین ورودی تست کرده و مشکلات آن را رفع کنید.

۶. مراحل ۴ و ۵ را برای هر عملیات تکرار کنید.

**نکته:** هرگز یک برنامه بزرگ را به طور کامل نوشته و بعداً کامپایل و تست نکنید، بلکه آن را به بخش‌های کوچک‌تری بشکنید که بتوان در مراحل متعددی برنامه را کامپایل و تست کرد.

**نکته:** همواره در تست کردن برنامه خود به این موضوع توجه کنید که کاربر می‌تواند هر ورودی (مجموعه اعداد، کارکترها، نشانه‌ها و ...) را وارد کند. برنامه شما باید برای ورودی‌های مختلف، رفتار مناسبی از خود نشان دهد. مثلاً اگر کاربر باید عدد وارد کند ولی حرف وارد می‌کند، باید پیغام خطا برای او ظاهر شود که ورودی او معتبر نیست! توجه کنید که بررسی این موارد در انجام پروژه بسیار مهم است و در صورت رعایت نکردن از شما نمره کسر خواهد شد! برای پیاده‌سازی این مورد تابع `errorOccurred` را پیاده سازی کنید:

```
void errorOccurred(char type){
    if(type=='o'){
        printf("ERROR:: out of range input!! ");
    }else if ...
        //Add your code here!!!
}
```