

به نام خداوند علم و دانش



دانشکده مهندسی کامپیوتر

مبانی کامپیوتر و برنامه نویسی (سی پلاس پلاس)

تمرین چهارم (Nested Loops)

دکتر مرضیه داودآبادی

پاییز 1403

طراحان تمرین: آقایان سیدامیرحسین حسینی جبلی و پوریارحیمی و پارسا زرگر

- در صورت وجود هرگونه ابهام به طراح پیام دهید.
 - باتوجه به وجود تاخیر 10 روزه، امکان تاخیر تحت هیچ شرایطی امکان پذیر نیست.
 - انجام تمرین ها تک نفره می باشد.
 - زبان برنامه نویسی سی پلاس پلاس است.
 - موارد ارسال شده به صورت آنلاین تحویل گرفته خواهند شد.
 - ددلاین تمرین: 6 آذر ساعت 23:59
 - برای دیدن تست کیس های نمونه متنوع به کوئرای درس بپیوندید.
 - https://quera.org/course/add_to_course/course/18682
 - Password: fcp03
 - آیدی طراحان: @amir_jebbeli / @Ray_bod / @parsazargar
-



01. ماتریس 4×4 (رحیمی)

برنامه ای بنویسید که یک ماتریس 4×4 به عنوان ورودی دریافت کند مانند زیر:

1	2	3	4
5	6	7	8
1	2	3	5
6	7	5	2

اکنون می‌خواهیم از یوزر شماره ی یک ستون خاص را گرفته (0 تا 3)، و سپس با استفاده از حلقه های تو در تو مجموع آن ستون را محاسبه کند.

ورودی

یک ماتریس 4×4 و شماره ی یک ستون

خروجی

جمع عناصر آن ستون.



02. قطر اصلی / فرعی (رحیمی)

برنامه ای بنویسید که یک ماتریس 4×4 به عنوان ورودی دریافت کند مانند زیر:

1	2	3	4
5	6	7	8
1	2	3	5
6	7	5	2

اکنون میخواهیم عناصر قطر اصلی و فرعی ماتریس پرینت شوند.

ورودی

یک ماتریس 4×4

خروجی

پرینت شدن عناصر قطر اصلی و قطر فرعی ماتریس



03.bubble sort(زرگر)

برنامه ای بنویسید که با روش [bubble sort](#) یک رشته ۲۰ تایی عدد دریافت کند و به روش [binary search](#) یک عدد را در آن پیدا کند. (اگر یافت نشد خطا بدهد)

ورودی

20 عدد +1 عدد برای جست و جو

خروجی

مرتب شده ی 20 عدد به همراه اعلام نتیجه جست و جو

04.بسط تیلور(زرگر)

برنامه ای بنویسید که رادیکال یک عدد را با دقت چهار رقم اعشار خروجی بدهد، توجه: حق استفاده از هیچ تابع ریاضیاتی در این تمرین ندارید. (کتابخانه `cmath` و توابعی که `pow` یا `sqrt` یک عدد را نمایش خواهند داد)

راهنمایی 1: استفاده از [بسط تیلور](#) در این تمرین

راهنمایی 2: برای محاسبه عدد مربع کامل نزدیک به عدد انتخاب شده می توانید از تابعی که شرح آن در زیر آمده استفاده کنید.

```
int F_nearestSquare(int number, int rootB){  
  
    if(number<0){  
  
        cout << "Exception: not negative number!";  
  
        return 0;  
  
    }  
  
    if(number<=rootB*rootB)  
  
        return rootB;  
  
    return F_nearestSquare(number,rootB+1);  
  
    /*recursive function*/  
  
    //number is between square of rootB-1 (rootA) and rootB  
}  
  
int main(void) {  
  
    int number;  
  
    int near=0;  
  
    cin >> number;  
  
    near=F_nearestSquare(number,near);
```



```
// بسط تیلور و محاسبه دقیق عدد
```

```
long double final_answer;
```

```
//...
```

```
}//bekhatere stack overflow say konid az adade bozorg mesle 1450 ya 26252 test begirid.
```

```
Good luck :)
```

➤ **توجه مهم!** اعداد کوچکتر از 1000 به خاطر کسر های بزرگ سری تیلور خطای زیادی دارند فلذا اگر عددی کوچکتر از 1000 به شما داده شد آنرا ابتدا ضرب در 400 و در انتها تقسیم بر 20 کنید، اینکار از نظر محاسباتی هیچ تفاوتی ایجاد نمیکند اما از خطای شما می کاهد.

```
if(number<1000){  
  
number*=400;  
  
lowerthan1000=true;  
  
}  
  
.  
  
.  
  
.  
  
if(lowerthan1000){  
  
final_answer/=20;
```



}

ورودی

عدد

خروجی

جذر، ریشه 2 مثبت، آن

05. ماتریس (زرگر)

برنامه ای بنویسید که حاصل دترمینان ضرب یک ماتریس 2×3 را در یک ماتریس 3×2 نمایش دهد. دو ماتریس را دریافت کرده ماتریس نتیجه را نمایش داده و در نهایت نوشته شود:

x00 x01 . . .

x10 x11

.

.

.

det($m \times n$) is:



راهنمایی: $m(3*2)*n(2*3)$

ورودی

در ابتدا m دانه دانه از $x00$ به صورت سطری پر می شود سپس n

خروجی

ماتریس نتیجه ضرب به صورت دانه دانه نمایش داده می شود. دترمینان آن محاسبه شده و به نمایش در می آید.

06. پردازش توان (زرگر)

علی با دوستانش بحثش شده است که نمی توانیم برنامه ای را بسازیم که بدون مرحله به مرحله تعریف کردن جمع سپس تعریف کردن ضرب از روی جمع و سپس تعریف کردن توان از روی ضرب به توان برسیم فرض کنید او توانسته است جمع دو متغیر را در زبان برنامه نویسی خود تعریف کند. او را کمک کنید بدون ساختن ضرب (طبیعتاً تابعی به اسم ضرب نباید نشان داده شود) این پیاده سازی را انجام دهد.

اجازه استفاده از توابع مختلف و ساخت آن‌ها و همچنین استفاده از عملیات ضرب را ندارید



راهنمایی: تصور کنید اگر $a \times b$ را بخواهیم به جمع بنویسیم بایستی b بار a یا a با b با خود جمع شود.

حال اگر a^b داشته باشیم خواهد بود a بار b در خود ضرب شود که اگر a بار b در a ضرب کنیم خواهیم داشت b بار، a بار a با خود جمع شود. مطلوبست نگارش حلقه ای که خروجی لازم را به ما بدهد.

راهنمایی 2: از `temp` استفاده کنید.

ورودی

دو عدد که قرار است عدد اول به توان عدد دوم برسد.

خروجی

حاصل توان

07. بسی رنج (حسینی)

این سوال سخت است، اگر ابتدا موفق نشدید ناامید نشین، باید بسی رنج برد تا به گنج رسید ♥

میخواهیم برنامه ای عجیب بنویسیم که ابتدا یک رشته از یوزر دریافت کند، مانند رشته زیر:

salam man amir hastam

➤ نکته: این اطمینان را میدهیم که یوزر رشته های حداقل 2 کاراکتری وارد میکند

توجه: برای محاسبه ی طول یک رشته از تابع length استفاده میکنم، به مثال زیر توجه کنید.

```
string txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
  
cout << txt.length(); // 26
```

توجه: برای دریافت رشته ای که space ها را هم بتوان گرفت به جای استفاده از cin از

getline استفاده میکنیم مانند زیر:

```
string reshte;  
  
getline(cin, reshte);
```

اکنون میخواهیم، از چپ رشته را بررسی کنیم و مثلاً وقتی حرف S را میبینیم، به تعداد S های موجود در رشته S های به هم چسبیده پرینت کنیم، بعد که a میبینیم، به تعداد a های موجود در رشته a های به هم چسبیده پرینت کنیم، بعد که کاراکتر | میبینیم، به تعداد | های موجود در رشته بالای، اهای به هم چسبیده پرینت کنیم، وقتی دوباره برای بار دوم a را میبینیم، چون قبلاً به تعداد کل a ها، a های به هم چسبیده پرینت شده بود، پس این بار ایگنور میشود و...

برای مثال خروجی مطلوب برای ورودی بالا خروجی زیر است:

```
ssaaaaaallmmmm nirht
```

فرض کنید رشته ورودی `salam` است.

➤ راهنمایی: برای حل این سوال از دو حلقه `for` باید استفاده کنید حلقه ی اول وظیفه ی

پیمایش کاراکترهای رشته را دارد، یعنی، `s` و `a` و `a` و `m` /// حلقه دوم هم رشته را از همان

جایگاه فعلی حلقه قبلی دنبال میکند و کاراکترها را با هم مقایسه میکند /// برای مثال،

اول `s` با `s,a,l,a,m` مقایسه میشود و هرکدام برابر با `s` بود، حرف `s` با رشته تهی اولیه ی

ما کانکت میشود، سپس به `a` میرسیم و `a` با `a,l,a,m` مقایسه میشود و هرجا برابر بودند

باز کانکت میشود و سپس `a` با `a,l,a,m` مقایسه میشود /// وقتی مجدد به `a` میرسیم، باید

یک مکانیسمی طراحی کرده باشید که این `a` چون قبلا مشاهده شده است، اکنون ایگنور

شود برای طراحی این مکانیزم کمی فکر کنید.

➤ توجه: اگر یک رشته تعریف کنیم و اسم آن را `str` بگذاریم؛

و اکنون بخواهیم به کاراکترهای آن رشته دسترسی داشته باشیم چه کار میکنیم؟ به کد زیر نگاه

کنید:

```
string str;  
  
str = "alireza";  
  
cout << str[2]; // i  
  
cout << str[3]; // r
```



```
cout << str[4]; // e
```

*توجه * : برای چسباندن یک کاراکتر به یک رشته چکار میکنیم؟ مثلاً میخواهیم کاراکتر R را به "ali" بچسبانیم. به تکه کد زیر نگاه کنید:

```
string str;  
  
str = "ali";  
  
str += 'R'; // out: "aliR"
```

ورودی

یک string

خروجی

در بالا توضیح داده شد.