یادآوری: تمامی تمرینات و اطلاعات مربوط به تحویل آنها بصورت هفتگی در سایت درس قرار داده میشوند:

http://ele.aut.ac.ir/~btaheri/cpp/

توجه: هنگام ارسال برنامه،کافیست فقط فایل .cpp رو ارسال کنید. (به هیچ وجه فایل .exe که فایل اجرایی است که توسط کامپایلر ساخته شده است را نفرستید!)

سوالات برنامه نویسی

۱ – **هدف:** کار با آرایه ها:

تابعی بنویسید که دو آرایه از اعداد **صحیح مرتب شده** (به همراه طول آنها) را **ورودی بگیرد**، و بدون استفاده از عمل مرتب سازی، آنها را در یک آرایه ی سوم، **بصورت مرتب شده** قرار دهد. برنامه ی خود را کامل کنید و نشان دهید که عملکرد آن درست است.

اطلاعات: تابع می تواند مشابه نمونه کد زیر باشد:

```
...
void mergeArray(int A[], int lengthA, int B[], int lengthB, int C)
{

   ...

}
int main()
{
   int firstLen, secLen;

   cout << "get number of the numbers at the first array: ";
   cin >> firstLen;
   cout << "get number of the numbers at the second array: ";
   cin >> secLen;

   int firstArray[firstLen], secondArray[secLen], mergedArray[secLen+firstLen];


   // get inputs:
   ...


   // merge:
   mergeArray(firstArray, firstLen , secondArray, secLen, mergedArray);

   // print array "mergedArray[]"
   . . .


}
```

خطر!: طول یک آرایه هیچگاه نمی تواند توسط یک متغیر تعیین شود! در صورتی که توسط یک متغیر طول آرایه تعیین شود نباید مقدار متغیر در ادامه ی برنامه عوض شود!!

سایز آرایه ها را باید به تابع ارجاع بدهیمتا بدانیم تا در داخل آن بدانیم هر کدام چقدر طول دارند!

۲ – **هدف:** کار با vector ها:

برنامه ی سوال قبل را دوباره با استفاده از وکتور ها بنویسید!

اطلاعات برنامه ی شما می تواند دارای شکل کلی زیر باشد:

```
. . .
vector <int> mergeArray(vector <int> A, vector <int> B)
```

```
{
   vector <int> C;

   ...
   return C;

}
int main()
{

   vector <int> firstVec, secondVec, mergedVec;
   // I am sure that the operator will not enter more than 100 numbers !

   cout << "get number of the numbers at the first vector: ";
   cin >> firstLen;
   cout << "get number of the numbers at the second vector: ";
   cin >> secLen;

   // get inputs:
   ...


   // merge:
   mergeArray(firstVec, secondVec, mergedVec);

   // print vector "mergedVec"
   . . .


}
```

خطر: بر خلاف آرایه ها، چون وکتور ها دارای زیرتابع ()size هستند، لازم نیست طول آن را به تابع ارجاع دهیم!

خطر: برخلاف آرایه ها، ارجاع وکتور ها به توابع call by value است!!

خطر: بر خلاف آرایه ها، می توان وکتور را به عنوان خروجی تابع به کار برد!!

۳ – **هدف:** کار با string ها

تابعی بنویسید که یک رشته(string) و یک کاراکتر (char) را به عنوان ورودی بگیرد و اندیس n–امین وقوع کارکتر گرفته شده را در خروجی نمایش دهد.

INPUT:
Enter the character:
e

Enter the string:
Hi Dr. Taheri! How are you! Nice to meet you!

OUTPUT:
There are 5 "e"s in the string!

۴ – **هدف:** کار با string ها

تابعی بنویسید که با گرفتن یک جمله از ورودی، چاپ کند که بیشتر کلمات با چه کاراکتری شروع می شوند.

INPUT:
Enter the string:
The first semester at cpp class, was the best semester and class I ever had !! Oh I remember that TA!!

OUTPUT:
The most repeated first-character is "c" with 3 repetitions!

۵ – **هدف:** بی هدف!

برنامه ای بنویسید که توسط محاسبات نردبانی ک.م.م و ب.م.م دو عدد را محاسبه و حاصل را درخروجی چاپ کند.

اطلاعات: برای هرکدام یک تابع بنویسید!

۶ – **هدف:** کار با توابع بازگشتی

سوال قبل را با استفاده از توابع بازگشتی به صورت زیر پیاده سازی کنید:

$$\gcd(x,y) = \begin{cases} x & \text{if } y = 0 \\ \gcd(y, \text{remainder}(x,y)) & \text{if } x \geq y \text{ and } y > 0 \end{cases}$$

$$\text{lcm}(a,b) = \frac{|a \cdot b|}{\gcd(a,b)}.$$

اطلاعات:

بزرگترین مضرب مشترک: Greatest Common Devisor (GCD)

کوچکترین مقسوم علیه مشترک: Least Common Multiple (LCM)

۷ – **هدف:** کار با توابع:

برنامه ای بنویسیدکه دو عدد را از ورودی بگیرد و مشخص کند که آیا دو عدد نسبت به هم اول هستند یا نه.

یادآوری: دو عدد متباین بزرگترین مقسوم علیه مشترکشان ۱ است.

اطلاعات: برای این کار یک تابع بنویسید.

۸ – **هدف:** بی هدف!

عنوان: اعداد آرمسترانگ

به عددی که حاصل جمع توان ۳ ارقام آن عدد با خوش برابر باشد اعداد آرمسترانگ گویند:

$153 = 1^3+5^3+3^3$

برنامه ای بنویسید که یک عدد از ورودی بگیرد و تشخیص دهد که عدد اول است یاخیر!

اطلاعات: لازم نیست برای این کار تابع نویسید!

۹ – **هدف:** توابع بازگشتی

سری اکرمن (Acreman) به صورت زیر تعریف می شوند:

$$Acr(0,n) = n+1$$
$$Acr(m,0) = Acr(m-1,1); m>1$$
$$Acr(m,n) = Acr(m-1, Acr(m,n-1)); n>1, m>1$$

برنامه ای بنویسید که دو عدد صحیح از ورودی را بگیرد و $Acr(m.n)$ را به ازای آن اعداد چاپ کند.

۱۰ -**هدف:** کار با رشته ها و کاراکتر ها

برنامه ای بنویسید که یک رشته را به عنوان ورودی گرفته وحروف کوچک آن را به حروف بزرگ، و حروف بزرگ آن را به حروف کوچک تبدیل کند.

نمونه:

INPUT:
Hello how are you?!

Output:
hELLO HOW ARE YOU?!

۱۱ -مرتب سازی حبابی: مرتب سازی حبابی ساده ترین و بدترین نوع مرتب سازی است. در این مرتب سازی ابتدا تمام داده ها در یک آرایه قرار می گیرند. سپس برنامه هر خانه از آرایه را با خانه ی بعدی مقایسه، آنها را جابجا می کند. بعد از جابجایی به تعداد لازم، مرتب سازی اتمام می یابد.

سودوکد: سودوکد این مرتب سازی به صورت زیر است:

1. Input n
2. Input an array of values A[1], A[2], …, A[n]
3. Do the following n-1 times
   a. Let i = 1
   b. Do the following n-1 times:
      i. If A[i] > A[i+1], exchange A[i] and A[i+1]
      ii. Add 1 to i
4. Output A

مثال: فرض کنید می خواهیم دنبالهی 5 1 4 2 8 را با مرتب سازی حبابی مرتب کنیم. ترتیب زیر مراحل اجرای حلقه ی فوق را نشان می دهد.

First Pass:

(**5 1** 4 2 8) ⟶ (**1 5** 4 2 8), Here, algorithm compares the first two elements, and swaps them.
(1 **5 4** 2 8) ⟶ (1 **4 5** 2 8), Swap since 5 > 4
(1 4 **5 2** 8) ⟶ (1 4 **2 5** 8), Swap since 5 > 2
(1 4 2 **5 8**) ⟶ (1 4 2 **5 8**), Now, since these elements are already in order (8 > 5), algorithm does not swap them.

Second Pass:

(**1 4** 2 5 8) ⟶ (**1 4** 2 5 8)
(1 **4 2** 5 8) ⟶ (1 **2 4** 5 8), Swap since 4 > 2
(1 2 **4 5** 8) ⟶ (1 2 **4 5** 8)
(1 2 4 **5 8**) ⟶ (1 2 4 **5 8**)

Now, the array is already sorted, but our algorithm does not know if it is completed. The algorithm needs one **whole** pass without **any** swap to know it is sorted.

Third Pass:

(**1 2** 4 5 8) ⟶ (**1 2** 4 5 8)
(1 **2 4** 5 8) ⟶ (1 **2 4** 5 8)

(1 2 **4 5** 8) ⟶ (1 2 **4 5** 8)
(1 2 4 **5 8**) ⟶ (1 2 4 **5 8**)

سوال: برنامه ای بنویسید که یک آرایه از اعداد را بصورت حبابی مرتب کرده و چاپ کند.

اطلاعات: فرض کنید که دنباله ی اعداد مورد نظر برای مرتب سازی دارای طول ۱۰ و به صورت زیر است:

-1 -20 0 10 21 10 9 4 -3 11

برنامه ی شما می تواند دارای شکل کلی زیر باشد:

```
. . .
void bubbleSort(int array[], int length)
{
   ...
}

int main()
{
   int array[] = {-1, -20, 0, 10, 21, 10, 9, 4, -3, 11};

   // sort:
   bubbleSort(array, 10);

   // print the sorted array:
   ...
}
```

۱۲ -برنامه ی فوق را با استفاده از وکتور ها پیاده سازی کنید.

۱۳ -**هدف**: کار با structure ها

استراکچر زیر را در نظر بگیرید:

```
struct Student
{
   int studentNumber;
   float finalGrade;
};
```

`studentNumber` عددی ۹ رقمی می باشد که شمارهی دانشجویی هر فرد است و `finalGrade` نمرهی هر دانشجو در امتحان پایانی میباشد. شما باید برنامه ای بنویسید که وکتوری از استراکچر معرّفی شده را درست کرده و سپس مقادیری تصادفی در آن قرار دهد. (می تواند از ورودی مشخصات چند دانشجو را بگیرد). سپس آن را به یک تابع ارجاع و وکتور ورودی را بر اساس نمرهی افراد و به صورت نزولی، مرتب کند.

اطلاعات: می توانید از تابع برنامه ی قبل در اینجا استفاده کنید!

اطلاعات: برنامه ی شما می تواند به صورت زیر باشد:

```
vector <Student>  sort(vector <Student> stuList)
{
   Vector <Student> sorted;

   ...

   return sorted;
}

int main()
{
   vector <Student> stuList, sorted;
   Student tmpStu;
   int tmpInt;
   float tmpFl = 1;

   // input some information from input: the last student is the one with student number 0
   cout << "Enter a new student number: ";
   while(cin >> tmpInt)
   {
      cout << "Enter the score: ";
```

```
        cin >> tmpFl;

        tmp.studentNumber = tmpInt;
        tmp.finalGrade = tmpFl;

        stuList.push_back(tmpStu);

        cout << "Enter a new student number: ";
    }


    // sort:
    sorted = sort(stuList);

    // print the sorted vector:
    ...
}
```

۱۴ -**هدف: کار با استراکچر ها:**

برنامه ی قبل را (به جای وکتورها) با استفاده از آرایه ها بنویسید. (آرایه ای از متغیر های ساخته شده با استراکچر)

۱۵ -**(سوال امتیازی) هدف:** کار با رشته ها

عنوان: مُگاوِبات دکتر قشقاویوس! (Cowculation!)

دکتر باستان شناس قشقاویوس حین تحقیقات خود در بقایای خرابه های آتلانتیس پرده از یک راز مهم برداشت. وی طی مکاشفات خود آثار یک تمدن گاوی برخورد کرد که این آثار محاسبات ریاضیشان نشات می گرفت. در واقع صد ها لوح گاوی در نزدیکی یکی از کشتارگاه های شهر پیدا شد و پروفسور موفق به کشف رمز آنها شد. مساله ی بسیار مهمی که کشف شد این بود که اگرچه گاو ها نمی توانند با دست هایشان محاسبات ریاضی را انجام دهند، ولی این کار را روی پاهایشان انجام می دهند. جناب دکتر طی کشف دوم به این نتیجه رسیده اند که گاو ها در سیستم شمارش خود از ۴ نشانه برای اعداد استفاده می کرده اند که عبارتند از D و C وU و V و همچنین جدول زیر را از بین الواح نامبرده استخراج کرده اند:

Sum	V	U	C	D
V	V	U	C	D
U	U	C	D	V,U
C	C	D	V,U	U,U
D	D	V,U	U,U	C,U

نماد اول در جدول حاصل جمع را نشان می دهد و نماد دوم رقم نقلی را. مثلا در این جدول:

U+V = D
C+D=UU
DVVCU+CVUCU = UUVCVC

سوال: برای کمک به جناب دکتر قشقاویوس توابعی بنویسید که دو عدد در سیستم گاوی (یعنی با نماد های گاوی بصورت string) بگیرند و هر کدام از اعمال زیر را انجام دهند.

a) دو عدد را جمع کند.
b) دو عدد را ضرب کند. (از تابع قبل استفاده کنید.)
c) عدد اولی را به توان عدد اولی برساند. (از تابع قبل استفاده کنید.)