

## Template Functions

در این سوال میخواهیم توابع زیر را به صورتی بنویسیم که بتوانیم آن را برای انواع مختلف متغیر ها صدا بزنیم. برای اینکار باید از مفهوم Template استفاده کنید.

- تابع GetMax این تابع دو حالت دارد یا دو ورودی را گرفته و ورودی بزرگتر را باز می گرداند و یا یک آرایه را گرفته و بزرگترین عضو آن آرایه را باز می گرداند
- تابع swap دو ورودی داده شده را با هم جا به جا می کند
- تابع Find این تابع با گرفتن یک آرایه از نوع دلخواه و آنچه که می خواهد پیدا کند اندیس جایی که پیدا شده را باز می گرداند و در غیر این صورت 1- باز می گرداند. ( دقت شود سائز آرایه را نمی تواند به عنوان ورودی تابع بدهید و باید از ویژگی های Template استفاده کنید)
- تابع Sort که با گرفتن یک آرایه آن را مرتب می کند اما این سورت بدین صورت است که :

۱. اگر جنس ورودی شما از نوع عددی بود ( int,float,double,long long int ) آرایه را می بایست از کوچک به بزرگ سورت کنید

۲. و اگر جنس داده های ورودی به صورت ( char , string ) بود از بزرگ به کوچک (z-a) سورت کنید

- تابع ValArray که با گرفتن یک ورودی آرایه ای از جنس آن ساخته و همه ی خانه هایش را با آن پر می کند و اشاره گری به آن را باز می گرداند به عنوان مثال :

```
valarray<int, 10>(17);
```

Copy

آرایه ای از نوع int به سائز 10 ساخته و تمام خانه هایش را با عدد 17 پر می کند.

- تابع Add که با گرفتن دو ورودی آن ها را با هم جمع می کند ( دقت شود در صورتی این ورودی از نوع char\* باشد باید یک char\* دیگر گرفته و دو ورودی را به هم متصل کنید. در این حالت باید برای این نوع ورودی تابعتون رو specialize بکنید)
- تابع Print : تنها کافیسست ورودی داده شده را چاپ نمایید: فقط در صورتی که این ورودی از نوع double باشد آن را به صورت نماد علمی چاپ کنید

## شبیه سازی کلاس وکتور

در این سوال قصد داریم کلاس vector را شبیه سازی نماییم.

میدانیم که vector در واقع نوعی ساختمان داده خطی است که در زبان c++ در قالب کتاب خانه ای با همین نام ارائه شده است و از تمام انواع داده های اصلی پشتیبانی میکند.

حال میخواهیم با توجه به مفهوم template class، کلاس vector را به گونه ای طراحی کنیم که از تمام انواع داده ای اصلی پشتیبانی نماید.

کلاس طراحی شده توسط شما میتواند برای پشتیبانی از یک ساختمان داده خطی، از آرایه به عنوان data member استفاده نماید.

و این کلاس باید قابلیت کپی کردن را دارا باشد (یعنی پیاده سازی کامل copy constructor) به گونه ای که با مساوی قرار دادن آبجکت های این کلاس در شرایط مختلف کلاس به درستی کار کند و ظرفیت و المان های موجود به روز شوند

از طرفی باید توابع زیر را به عنوان function member ها پیاده سازی نمایید:

```
void push_back(T data);
```

- وظیفه این داده اضافه کردن data به انتهای vector میباشد.

```
void push_front(T data);
```

- وظیفه این تابع اضافه کردن data به ابتدای vector است.

```
void insert(int index, T data);
```

- این تابع عنصر data را در محل گفته شده (index) اضافه می نماید.

```
void pop();
```

- این تابع آخرین عنصر موجود در vector را در صورت وجود حذف می نماید.

```
void delete(int index);
```

- وظیفه این تابع حذف عنصر از محل داده شده (index) است.

```
int search(T data);
```

- هدف از این تابع جست و جوی مقدار data در vector و برگرداندن اندیس اولین مورد یافت شده می باشد. (در صورتی که عنصر مورد نظر یافت نشود، مقدار -1 برگردانده شود.)

```
int size();
```

- وظیفه این تابع برگرداندن سایز vector می باشد.

```
void swap(int index1, int index2);
```

- این تابع محتوای موجود در مکان های index1 و index2 را جا به جا می نماید.

```
void reverse();
```

- هدف از پیاده سازی این تابع برعکس کردن محتوای vector می باشد.

```
void sort();
```

- از این تابع برای مرتب سازی صعودی آرایه استفاده میشود.

**توجه: برای کد خود رابط کاربری مناسب بنویسید.**

## template template template

فرض کنید یک کلاس Template به نام Storage که در پایین implement شده است داریم.

```
template <class T>
class Storage
{
private:
    T m_value;
public:
    Storage(T value)
    {
        m_value = value;
    }

    ~Storage()
    {
    }

    void print()
    {
        std::cout << m_value << '\n';
    }
};
```

همانطوری که مشخص است این کلاس کارش ذخیره سازی داده از نوع های متفاوت می باشد. حال سوال اینجاست آیا واقعا برای تمام انواع متغیر به صورت درست عمل می کند؟ فرض کنید این کلاس را بخواهیم برای نوع pointer ای استفاده کنیم آیا همچنان اگر تابع print را صدا بزنیم به صورت درست کار می کند؟ اگر درست کار نمی کند پیشنهاد شما برای درست کردنش چیست؟ ( به عنوان مثال اگر بخواهیم برای \*int استفاده کنیم. منظور از \*int آرایه ای از integer ها نیست و تنها به پوینتر مدنظرمون هستش به عنوان مثال کد زیر را در نظر بگیرید) دقت کنید فقط از نوع \*int نیست و انواع پوینتر های مختلف مثل \*float و ... نیز مدنظرمون می باشد.

```
int main()
{
```

```

Storage<int> myint(5);
myint.print();

int x = 7;
Storage<int*> myintptr(&x);

x = 9;
myintptr.print();

return 0;
}

```

انتظار داریم پس از اجرای کد بالا عدد های 5 و 7 چاپ شوند.

حال اگر بخواهیم برای نوع متغیر `*char` که درواقع آرایه ای `char` ها می باشد از این کلاس استفاده کنیم باید چه کنیم؟

```

int main()
{

    char* string = new char[40];

    std::cout << "Enter your name: ";
    std::cin >> string;

    Storage<char*> storage(string);

    delete[] string;

    storage.print();
}

```

پس در نهایت کدتون به عنوان مثال باید برای همچین `main` ای کار بکند.

```

int main()
{

```

```
Storage<int> myint(5);
myint.print();

double x = 7.5;
Storage<double*> myintptr(&x);

x = 9;
myintptr.print();

char *name = new char[40];
strcpy(name, "Alex");

Storage< char*> myname(name);

delete[] name;

myname.print();
}
```

## (اختیاری) کلاس گراف

متن سوال را از pdf کناری دریافت کنید.