# UoA CP Lecture 1
# Shifting to C++(or C with STL)

Qingchuan(Sam) Zhang

April 3, 2020

## Contents

# 1 Background

In competitive programming, no expertise in C++/Java/Python is required. We only need to a small portion of the language(s) to convert our ideas to codes. Here I'll introduce the basic stuff needed for CP. Though there are more advanced stuff might be useful, the ones here can fulfill our needs for now.

The documentation of C++ is on cppreference. Use the docs to find things you don't know how to use.

Let's start!

## 2 Basic Framework

```cpp
#include <iostream>
#include <string>
#include <algorithm>
#include <vector>
#include <queue>
// or #include<bits/stdc++.h> if supported
using namespace std;

int main() {

    return 0;
}
```

**`#include`** This is like `import` in Python, you need to include the libraries before using built-in functionalities.

**`using namespace std`** This line means you want to use the functions/containers residing in the `std` namespace as if they were written by yourself. If you don't put this line, you will need to add `std::` before the names. For example, `sort` versus `std::sort`

**`main`** This is where your program starts to run. The return type should always be `int` and it always return 0.

# 3 Basic Data Types

- `bool`: either `true` or `false`.
- `char`: an ASCII character, usually used to store `'a'-'z'` or `'A'-'Z'` or `'0'-'9'`.
- `int`: integers range from $-2^{31}$ to $2^{31}-1$, or just memorize $\pm 10^9$
- `long long`: integers range from $-2^{63}$ to $2^{63}-1$, or just memorize $\pm 10^{18}$
- `double`: decimal numbers of 15 digit precision.
- `std::string`: stores a sequence of characters like `"abcdefg"` or`"from1to2"`

# 4 Input/Output

## 4.1 C++ style(recommended)

```cpp
#include<iostream>
using namespace std;
int main(){
    int a;
    long long b;
    double c;
    cin >> a >> b >>c;
    cout << a << " " << b << " " << c << endl;//endl = end of line
    return 0;
}
```

## 4.2 C style(not recommended)

```cpp
#include<cstdio>
int main(){
    int a;
    long long b;
    double c;
    scanf("%d %lld %lf",&a,&b,&c);
    printf("%d %lld %f",a,b,c);
    return 0;
}
```

# 5 if/for/while

The following three structures all the most common ones used by almost every single c++ program.

## 5.1 if

```
//...
int main(){
    int value;
    cin>>value;
    if(value == 0){
        cout<<"the input is zero"<<endl;
    }else{
        cout<<"the input is not zero"<<endl;
    }
    return 0;
}
```

## 5.2 for

```
//...
int main(){
    int n;
    cin>>n;
    int sum = 0;
    for(int i = 1;i<=n;i++){
        sum = sum + n;
        //sum += n;
    }
    cout<<"The sum from"<<1<<" to n is "<<sum<<endl;
    return 0;
}
```

## 5.3 while

```
//...
int main(){
    int value;
    cin>>value;
    int step = 0;
    //collatz conjecture
    while(value != 1){
        if(value % 2 == 0){
            value /= 2;
        }else{
```

```
11              value = value * 3 + 1;
12          }
13          step++;
14          //step += 1;
15      }
16      cout<<"The number of steps is "<<step<<endl;
17      return 0;
18  }
```

# 6 Array

## 6.1 std::vector<Type>

```cpp
//..
#include<vector>
int main(){
    vector<int> a(5,1);//{1,1,1,1,1}
    a.insert(a.begin()+1,2);//{1,2,1,1,1,1}
    //0-indexed
    for(int i = 1;i<a.size();i++){
        a[i] = a[i-1];
    }
    int last = a.back();
    bool is_empty = a.empty();


    return 0;
}
```

## 6.2 Type a[]

```cpp
//...

int a[100]; //{0,0,0,...}
const int maxn = 200;
int b[maxn];
int n;
int c[n];//error
int main(){
    int d[100];// random values
    int value = d[50];
    cout<<value<<endl;
    return 0;
}
```

# 7 Example Problem: Increasing Array

```cpp
#include<bits/stdc++.h>
using namespace std;
const int MAXN=200000+5;
int n;
int nums[MAXN];

int main()
{
    cin>>n;
    for(int i=0; i<n; i++)
        cin>>nums[i];
    long long ans=0;
    for(int i=1; i<n; i++)
    {
        if(nums[i]>=nums[i-1])
            continue;
        else
        {
            ans+=nums[i-1]-nums[i];
            nums[i]=nums[i-1];
        }
    }
    cout<<ans<<endl;
    return 0;
}
```

# 8 String

## 8.1 std::string

```cpp
//..
#include<string>
int main(){
    string a = "abcdefg";// set to literal value
    string b; // set to ""
    string c;
    cin>>c; // input
    b = "xyz";
    c = c + a + b;// concatenation
    cout<<c<<" "<<c[0]<<endl;
    return 0;
}
```

## 8.2 char s[]

```cpp
//...
int main(){
    char s[5] = "abcd";//"abcd\0"
    char a[10];
    scanf("%s",a);
    printf("%s",a);
    return 0;
}
```

# 9 Struct keyword

## 9.1 Definition

```
struct Point{
    int x,y;
    Point(int xx,int yy){
        x = xx;
        y = yy;
    }
    double disFromOrigin(){
        return sqrt(x*x + y*y);
    }
};

int main(){
    Point p(10,20);
    cout<< p.disFromOrigin() << endl;
    return 0;
}
```

## 9.2 Example: Codeforces 479 C

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int MAXN = 5000 + 10;
int n;
struct exam_time
{
    int atime, btime;
    exam_time(){};
    exam_time(int a, int b)
    {
        this->atime = a;
        this->btime = b;
    }
    bool operator<(const exam_time &exam)
    {
        return atime < exam.atime || (atime == exam.atime && btime < exam.btime);
    }
} exam[MAXN];
int main()
{
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        int x, y;
        cin >> x >> y;
        exam[i] = exam_time(x, y);
    }
    sort(exam, exam + n);
    int cur = 0;
    for (int i = 0; i < n; i++)
    {
        if (exam[i].btime < cur)
        {
            cur = exam[i].atime;
        }
        else
        {
            cur = exam[i].btime;
        }
    }
    cout << cur << endl;
    return 0;
}
```