**A little sytax needworthy**

* Cin:
* Cin.fail(); : Kiểm tra xem Cin có bị sai không.
* Cin.clear(); : Xóa error flag trong bộ nhớ đệm.
* Cin.ignore(number,’char’); : Xóa ký tự trong bộ nhớ đệm.(number ký tự, tới ký tự char đầu tiên).//nhớ có getline thì cin xong phải xóa ngay.
* Stop:
* Break : Kết thúc vòng lặp ngay lập tức khi gặp.
* Continue: Dùng để nhảy đến cuối vòng lặp hiện tại, bỏ qua tất cả mọi lệnh dưới continue.

\*Notes: Thường dùng ở vòng lặp for, cẩn thận khi dùng ở while và Do-while vì nếu nhảy qua câu để nâng cấp điều kiện thì loop sẽ thành vô tận.

* Exit(): Dừng lại cả chương trình.
* Random number:

1. Phương pháp cũ(k ưu tiên):

#include<ctsdlib>; //for rand() and srand()

#include<time>; //for time()

Srand(time(null)); //or 0

Int a= rand();

//rand\_max= full int;

Cout<<a;

* Cách giới hạn chữ số :

Rand%100 + 1 -> giới hạn dưới 100.(từ 1->100)

Rand%1000+1 -> giới hạn dưới 1000.

…..

1. Phương pháp mới (ưu tiên):

#include<random>;

Random\_device <name random>;

Mt19937 <name(name random())>;

Auto n= <name>; // khai báo, định nghĩa.

* Cách giới hạn chữ số:

Uniform\_int\_distribution<int> <uniname(n\_min,n\_max)>;

Auto n=<uniname(name)>;

* Test data type:
* Typeid(variable).name().
* Convert in the datatype:

1. Str-> char:

* Char \*C=(char\*)str.c\_str();

1. Str-int:

* Chuyển qua char rồi dùng hàm atoi():

Int num=atoi(char[]);

* Sử dụng hàm stoi(str):

Int num=stoi(str); đơn giản.

1. Chuyển từ int sang char:

Int I=123;

* Ta có thể sử dụng : c=(char) I;

1. Chuyển từ int sang str:

* To\_string(num);
* Continue, break, exit, return:

1. Continue:

* Dùng phát là nó bỏ qua các câu lệnh ở dưới nó luôn.
* Sử dụng với vòng lặp cuối cùng bao nó.

1. Break:

* Dùng phát là tắt cả vòng lặp.
* Không break hàm được.

1. Exit():

* Dùng cái là tắt chương trình.
* Dùng với hàm thì khi gọi hàm nó tắt chương trình luôn.

1. Return <value>:

* Hàm gặp được return thì tự động đóng hàm, trả về giá trị cho hàm.
* Algorithm:

#include<algorithm>;

1. Min, max function:

Min(a,b,c); // find the min value in ele a,b,c…

Max(a,b,c); //max

1. Min, max in array:

Int a[3]={1,2,3};

Syntax: min\_element(a, a+3);

* Return the pointer point to address have min value.

With vector : min\_element(begin, end);

* Return the iterator to address have min value.

Because it return the iterator or pointer, then we need use

Operator dereference (\*) to take a value of function.

Similar with max\_element.

1. Accumulate (sum):

Int a[5]={1,2,3,4,5};

Int sum= accumulate(a, a+5);// adjust about (a+x, a+y+1) to count sum into about (x,y);

With vector, we use with iterator.

1. Swap: swap(a,b);
2. Sort: (O(nlogn)) // it’s faster with bubble arrangement ☺

Int a[5]={1,4,2,5,7};

Sort(a, a+5); // count into about like accumulate

// the default at function is increase. If we want to arrange decrease, we add “greater<int>”

// ex: sort(a, a+5, greater<int>

With vector as above.

If we want to sort follow our quality we can:

Use cmp: sort(a, a+n, cmp)

With “cmp” is a function to define the quality we need.

Ex: bool cmp(int a, int b)

{

Return a>b;

}

Stable\_sort(….); it’s stable with position.(n^2)

1. Find: find(a, a+5, “element”);// find ele in arr or vector.

// return pointer. If not not found, return pointer in a+5.

// with vector as above.

//O(N)

1. Binary\_serch(a, a+5,”element”);

// only use with arr have been order.

// O(logn)

1. Pow(x,i)=x^i.

* Tips:
* If the question not have limited to cin. We can use code:

String s;

While (cin>>s)

{}

And when we live in console screen, we can enter unlimited text and we can end with keystrokes Cirl-Z + Enter.

* Setw()<<right: when we enter input, input will insert from right to left else with left. Cout<<Setw()<<setfill(‘char’): change the char of setw.

Vd: setw(5)<<right<<setfill(‘-‘)<<’a’;

Out: ----a

Or : setw(5)<<left<<setfill(‘-’)<<’a’;

Out: a----

* How to use #include connecting the file:
* Syntax: #include “file.cpp”.
* Note\*: when we create many file cpp or header that need to include, if code into this file has class or structure, it will happen bug, because class and structure will repeat when program compile and execute.
* Solution: we declared “#pragma once” above class or structure that we know that it will be repeat.