C++

Introduction Challenges

1.Hello, World! Score: 5

#include <iostream>

#include <cstdio>

using namespace std;

int main() {

printf("Hello, World!");

return 0;

}

2. Input and Output Score: 5

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

int a,b,c;

cin >> a >> b >> c;

cout<< a+b+c;

return 0;

}

3. Basic Data Types Score: 10

#include <iostream>

#include <cstdio>

using namespace std;

int main() {

// Complete the code.

int i=0;

long int li=0;

long long int ll=0;

char c=' ';

float f=0.0;

double fl=0.0;

int s;

s=scanf("%d %ld %lld %c %f %lf",&i,&li,&ll,&c,&f,&fl);

printf("%d\n%ld\n%lld\n%c\n%.2f\n%.5lf",i,li,ll,c,f,fl);

return 0;

}

4. Conditional Statements Score:

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

int n=0;

cin >> n;

switch(n) {

case 1: cout << "one";

break;

case 2: cout << "two";

break;

case 3: cout << "three";

break;

case 4: cout << "four";

break;

case 5: cout << "five";

break;

case 6: cout << "six";

break;

case 7: cout << "seven";

break;

case 8: cout << "eight";

break;

case 9: cout << "nine";

break;

default: cout << "Greater than 9";

}

return 0;

}

5. For loop Score: 10

#include <iostream>

#include <cstdio>

using namespace std;

int main() {

// Complete the code.

int a,b;

cin >> a;

cin >> b;

for(int n = a; n <= b; n++ ) {

if( (n)>=1 && (n)<=b) {

switch(n) {

case 1: cout << "one\n";

break;

case 2: cout << "two\n";

break;

case 3: cout << "three\n";

break;

case 4: cout << "four\n";

break;

case 5: cout << "five\n";

break;

case 6: cout << "six\n";

break;

case 7: cout << "seven\n";

break;

case 8: cout << "eight\n";

break;

case 9: cout << "nine\n";

break;

default: if((n)%2==0) {

cout << "even\n";

} else {

cout << "odd\n";

}

}

}

}

return 0;

}

6. Functions

#include <iostream>

#include <cstdio>

using namespace std;

/\*

Add `int max\_of\_four(int a, int b, int c, int d)` here.

\*/

int max(int a,int b) {

if(a <= b) {

return b;

} else {

return a;

}

}

int max\_of\_four(int a, int b, int c, int d) {

int max1, max2;

max1 = max(a,b);

max2 = max(c,d);

return max(max1,max2);

}

int main() {

int a, b, c, d,s;

s=scanf("%d %d %d %d", &a, &b, &c, &d);

int ans = max\_of\_four(a, b, c, d);

printf("%d", ans);

return 0;

}

7. Pointer

#include <stdio.h>

void update(int \*a,int \*b) {

int sum, diff;

sum = (\*a)+(\*b);

if((\*a)>=(\*b)){

diff= (\*a)-(\*b);

} else {

diff=(\*b)-(\*a);

}

(\*a)=sum;

(\*b)=diff;

// Complete this function

}

int main() {

int a, b,s;

int \*pa = &a, \*pb = &b;

s=scanf("%d %d", &a, &b);

update(pa, pb);

printf("%d\n%d", a, b);

return 0;

}

8. Arrays Introduction

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

int n,a,s;

int arr[1000];

cin >> n;

for(int i=0;i<n;i++) {

s=scanf("%d",&a);

arr[i]=a;

}

for(int i=n-1;i>=0;i--)

printf("%d ",arr[i]);

return 0;

}

Score: 25

9. Operator Overloading

class Matrix {

public:

vector<vector<int>> a;

//Matrix();

Matrix operator+(const Matrix& right){

Matrix sum = \*this;

int n = right.a.size();

int m = right.a[0].size();

for(int i = 0; i<n;i++){

for( int j = 0; j<m; j++){

sum.a[i][j]=a[i][j]+right.a[i][j];

}

}

return sum;

}

};

Score: 30

10. Variable Sized Array

Int main() {

int s;

unsigned int N=0,Q=0,slength=0,a=0,b=1;

cin >> N >> Q;

unsigned int\*\* seq = new unsigned int\*[N];

unsigned int\*\* query = new unsigned int\*[Q];

int i = 0;

int j = 0;

for( i=0; i<N;i++){

cin >> slength;

seq[i]= new unsigned int[slength];

for(j = 0; j<slength; j++){

cin >> seq[i][j];

}

}

for( i=0; i<Q;i++){

query[i] = new unsigned int[2];

cin >> query[i][a] >> query[i][b];

}

for( i=0; i<Q;i++) {

a = query[i][0];

b = query[i][1];

cout << seq[a][b];

cout << endl;

}

//free dynamic space in the heap

for(i = 0; i<N;i++) {

delete [] seq[i];

}

delete [] seq;

for( i=0; i<Q;i++){

delete [] query[i];

}

delete [] query;

return 0;

}

Strings Challenges

Score:10

1.Strings

#include <iostream>

#include <string>

using namespace std;

int main() {

// Complete the program

string s1,s2;

char c;

cin >> s1 >> s2;

cout << s1.size() << " " << s2.size() << endl;

cout << s1+s2 << endl;

c=s1[0];

s1[0]=s2[0];

s2[0]=c;

cout << s1 << " " << s2;

return 0;

}

2.StringStream

#include <sstream>

#include <vector>

#include <iostream>

using namespace std;

vector<int> parseInts(string str) {

// Complete this function

char c = 'c';

vector<int> arr;

stringstream ss(str);

int num;

while( ss >> num >> c)

arr.push\_back(num);

ss >> num;

arr.push\_back(num);

return arr;

}

int main() {

string str;

cin >> str;

vector<int> integers = parseInts(str);

for(int i = 0; i < integers.size(); i++) {

cout << integers[i] << "\n";

}

return 0;

}