Test One V1:

Name:

Surname:

Student Seneca Email ID:

Q1: [10 marks]  
There is a treasure chest hidden in a room at the end of a path in magic mansion. You are to look for this hidden treasure and find your way back to the entrance of the mansion. Each door of this magic mansion opens into a different room with many doors with unique colors (the color of the doors in each room is unique), but number of doors are different in each room. To get to this treasure you need to check every and each door that opens to another room with many doors. The problem is that, this way you will easily lose your way back will be lost in the magic mansion for rest of your life. Luckily you are a programmer and you can write a program to help you with this.

To keep track of the way back and be able to check every and each room without being lost, write a Stack. (complete with header file and cpp file, with all necessary includes and etc…)

This Stack is capable of keeping color of the doors as its data. So you can push the color of the door when you enter a new room, and pop a color, to know which door leads you back to the room you came from.

The stack should have all the standard features: constructor, destructor, push, pop, isEmpty.

The push, pushes name of a color into the stack.

The pop, pops a color out of the stack and sends it back (it is your choice to see if it should return or copy the data back through the function argument)

isEmpty becomes true or false, depending on the stack being empty or not.

Q2: [ 10 marks]  
Write a program as a user interface to use the above stack: (complete with all header files included and etc…)

Step one: Create a menu function that returns an integer and receives nothing.

The menu function should display the following:  
1 – Open door  
2 – Go Back   
3 – Found Treasure, Show the way out  
0 – Exit program

and wait for the user to enter an integer, if the number user enters is one the above, return the number and exit the function, if the number is invalid (not one the above in any way) print: “invalid entry, try again:” and wait for user to try again.

Step two:  
Write the main function that accepts your first name and last name as arguments on the command line.  
  
Create an instance of the stack you created in Q1.   
If the program is compiled and executable is saved as “pathway”, the program should execute as follows if your first name and last name is “Fred Soley”:  
$ pathway Fred Soley <ENTER>  
and greet you as:  
Hello Fred Soley, welcome to the pathway program.

Then display the menu and wait for the user response and act accordingly and then redisplay the menu, until the user chooses to exit the program. Then quit the program by, printing:  
Goodbye Fred, I hope you have found the treasure  
and exit.

If the user selects “1” get a color name from the user and push it into the stack.  
If the user selects “2”, pop a color out of the stack and display it to the user.  
If the user selects “3”, pop all the colors one by one and display them all until the stack is empty.

Q3: [20 marks ] Determine the exact output of the following program and at any case if the exact output is not possible, write “undefined”.

#include <iostream>

#include <cstdarg>

using namespace std;

void w1(){ // size of an integer is 4 bytes

int a[3][100];

int\* p = (int\*)a;

cout<<"w1-----"<<endl;

cout<<sizeof(a)<<endl;

cout<<sizeof(p)<<endl;

}

void w2(){

int a[5] = {10, 20, 30, 40, 50};

int\* p = a;

cout<<"w2------"<<endl;

cout<<\*p++<<endl;

cout<<\*p<<endl;

cout<<(\*p)++<<endl;

cout<<\*p<<endl;

}

int& foo(){

static int i = 10;

return i;

}

int& foo(int& a){

return a;

}

void w3(){

int i = 30;

int& R = i;

int j = 50;

cout<<"w3------"<<endl;

cout<<R<<endl;

cout<<foo()<<endl;

foo() = 300;

cout<<foo()<<endl;

foo(j) = 400;

foo(R) = 500;

cout<<j<<endl;

cout<<i<<endl;

}

#define mk(v, t) t v

#define mkNst(var, t, val) t var = val

#define rep(n, i) for(i=0;i<n;i++)

#define prln(s) cout<<s<<endl

void w4(){

mk(i, int) = 2;

mkNst(b, int, 5);

prln("w4-------");

rep(b, i){

prln(i);

}

}

int S(int num, ...){

int s = 0;

int arg;

int i;

va\_list varg;

va\_start(varg, num);

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

arg = va\_arg(varg, int);

s += arg;

}

va\_end(varg);

return s;

}

void w5(){

int val;

cout<<"w5-------"<<endl;

val = S(4, 10, 20, 30, 40);

cout<<val<<endl;

val = S(4, 10, 20, 30, 40, 50, 60, 70);

cout<<val<<endl;

val = S(4, 10, 20, 30);

cout<<val<<endl;

}

int main(){

w1();

w2();

w3();

w4();

w5();

return 0;

}