MIDTERM 1 CS111 Summer 2020

NAME_____ CUNYID_____

Instructions:

1) Make sure your name and CUNY ID are filled in.

2) When asked to write a program, begin with the main portion of the program. In most cases I have written the beginning of the function for you and you just fill in the function body.

3) The only outside functions you can use are:
a) cout, cin, rand(), srand(), and time()
b) functions you write yourself
c) functions that appear anywhere on this test

SECTION 1 – Programming Questions 3 Points each

Q1) What are two reasons to write a program using a low-level language?

a) Speed
b) Small executable size

Q2) C++ has similarities to both low-level languages and high-level languages. What makes C++ similar to high level languages?

a) Objects

Q3) Why is the highest value that an integer can hold is 2147483647 and the highest value that a char can hold is 255?

a) Because of the variable size 1 byte (char) versus 4 bytes (int)

Q4) Why are programs used for stock trading systems frequently written in C++ as opposed to Java?

a) C++ is faster because its executable is optimized machine language.

Q5) Provide two ways that a void function can affect other functions?

a) Passing by reference
b) globals

Q6) What is the difference between passing a variable by value or by reference?
Whether the variable in the calling function gets modified.

Q7) What is the name of the function that when it exits the program stops execution?
main()

Q8) What is the difference between a global and local variable?
global variables retain their value after the function exits or scope containing variable ends

Q9) In the code below, which variable(s) are global to the program?
i and j

```
int i;
int j;
int main()
{
int global;
{
double d;
}
}
```

Q10) What is the name of the program that starts up a computer when it is turned on?

BIOS – Basic Input Output System

Section 2 – Understanding Programming Logic 5 Points Each

In this section, create the output requested using all the variable(s) and the values provided below.

For example:

Given int i = 2, j= 2, k=10 output 2

Answer: cout << k/(i+j)
Q1) Given int i = 3, j= 3, k=4 output 5?
```
cout << (i/j)+k
```

Q2) Given int i = 2, j= 2, k=10 output 0?
```
cout << k*(j-i)
```

Q3) Given int i = 5, j= 5, k=5 output 5?
```
cout << i * j / k
```

Q4) Provide code that outputs a random number 10 through 20?
```
srand(time(0));
cout << 10 + rand() %11 << endl;
```

Q5) Provide code that outputs a random character ‘a’ through ‘y’?
```
srand(time(0));
char c = ‘a’+ rand() %25;
cout << c << endl;
```

Q6) Using char c = ‘A’; Output “BCD” without changing the value of c.
```
char c1 = c + 1;
char c2 = c + 2;
char c3 = c + 3;
cout << c1 << c2 << c3 << endl;
```

Section 3 – Programming 10 Points each

Q1) RandomChar returns a random character
between 'A' .. 'Z','a'..'z', or '0'..'9'
Example: RandomChar() ==> 'S'
Example: RandomChar() ==> 'a'
Example: RandomChar() ==> '5'

```
char RandomChar()
{
        srand(time(0));
        int type = rand() % 3;
        if (type == 0)
                return('A' + rand() % 26);
        else if (type == 1)
                return('a' + rand() % 26);
        else
                return('0' + rand() % 10);
}
```

Q2) **RoundUp** takes a double and rounds up or down to the closest integer.
This can be done in one line of code!
Examples:

RoundUp(4.000) ➔ 4
RoundUp(4.4999) ➔ 4
RoundUp(4.51111) ➔ 5

```
int RoundUp(double d)
{
	return(d + .5);
}
```

Q3) SumOf3Digits2 takes an integer of three digits and returns their sum
SumOf3Digits2(541) ➔ 10 (5 + 4 + 1)
SumOf3Digits2(123) ➔ 6 (1 + 2 + 3)
SumOf3Digits2(100) ➔ 1 (1 + 0 + 0)

```
int SumOf3Digits2(int i)
{
	int i1 = i % 10;
	i /= 10;
	int i2 = i % 10;
	i /= 10;
	int i3 = i % 10;
	return(i3 + i2 + i1);
}
```

Q4) Max returns the largest value using as few lines as possible.
Sample Output: Max(0, 1, 0) ➔ 1
Sample Output: Max(0, 0, 0) ➔ 0
Sample Output: Max(-1, -2, -3) ➔ -1

```
int max(int i, int j)
{
	if (i > j)
		return(i);
	else
		return(j);
}

int Max(int i, int j, int k)
{
	return(max(i, max(j, k)));
}
```

Q5)
Create the functions **AskTheUserForInput(),GetUserInput, and HighestNumber** that are used in the program below.

```
void AskTheUserForInput()
{
	cout << "Please enter two numbers" << endl;
}

void GetUserInput(int& i1, int& i2)
{
	cin >> i1;
	cin >> i2;
}

int HighestNumber(int i1, int i2)
{
	return(max(i1, i2));
}

int main()
{
	// we will ask the user for two numbers and return
	// the larger of the two
	int i, j;
	//ask the user for two whole numbers
	AskTheUserForInput();

	GetUserInput(i, j);

	cout << "The highest number is " << HighestNumber(i, j) << endl;

	return(0);
}
```

Q6)

Mod Without using the mod function (%) write the mod function below

Example Mod(8, 8) ➔ 0

Example Mod(7, 5) ➔ 2

Example Mod(5, 7) ➔ 5

```
int Mod(int a, int b)
{
	return(a - b*(a / b));
}
```

Q7)

Finish the program below that asks the user for two numbers and the operation they want to perform. The valid operations are '+' and '-'. If the user doesn't enter a valid operation, the program should exit with an error message, otherwise the requested operation should be output.

Sample **Input Output**

'2' '5' '+' 7

'2' '5' '-' -3

'2' '5' '*' "In valid Operation. Run program again"

```
int main()
{
cout << "Enter two whole numbers." << endl;
int num1, num2;
cin >> num1;
cin >> num2;
cout << "Enter the operation you wish to perform on these numbers."
<< endl;
cout << "Choices are one of the following: +,-" << endl;
char operation;
cin >> operation;

if(operation != '+' && operation != '-')
{
        cout << "Invalid Operation. Run program again"  << endl;
        return(-1);
}

if(operation == '+')
        cout << num1 + num2 << endl;
else
        cout << num1 - num2 << endl;


return(0);
}
```

Q8) SumOf3Digits1 takes a double of three digits and returns their sum

SumOf3Digits1(5.41) ➒ 10 (5 + 4 + 1)

SumOf3Digits1(12.3) ➒ 6 (1 + 2 + 3)

SumOf3Digits1(1.00) ➒ 1 (1 + 0 + 0)

```
int SumOf3Digits1(double d)
{
        int i = d;
        if(i  != d)
                d *=10;
        i = d;
        if(i  != d)
                d *=10;
        i = d;
        if(i  != d)
```

```
            d *=10;
        i = d;
        return((i%10)+(i/10)%10+(i/100));
}
```

Q9) max returns the largest value passed in
Sample Output: max(0, 1) ➔ 1

Sample Output: max(0, 0) ➔0
Sample Output: max(-1,-2) ➔ -1

```
int max(int i, int j)
{
	if (i > j)
		return(i);
	else
		return(j);
}
```

Q10) Power returns d to the power of pow where pow can be 0..3
Power(6.3,0)==> 1
Power(6.3,1)==> 6.3
Power(6.3,2)==> 39.69
Power(6.3,3)==> 250.047
Power(6.3)==> 39.69

```
int Power(double d, int pow = 2)
{
	if (pow == 0)
		return(1);
	else if (pow == 1)
		return(d);
	else if (pow == 2)
		return(d * d);
	else
		return(d * d * d);
	return(0);
}
```

Q11)
SumOfTwoHighest returns the sum of the two largest numbers
SumOfTwoHighest(1,2,3) ➔ 5
SumOfTwoHighest(0,1,0) ➔ 1
SumOfTwoHighest(-1,-2,-3) ➔ -5

```
void swap(int& i1, int& i2)
{
	int temp = i1;
	i1 = i2;
	i2 = temp;
}

int SumOfTwoHighest(int i, int j, int k)
{
	if (i > j)
		swap(i, j);
```

```
        if (j > k)
                swap(j, k);
        if (i > j)
                swap(i, j);
        return(j + k);
}
```

Q12)

PrintThreeNumbers - helper function prints out i. If zeroFill is true then pads the number with up to 2 zeroes

Example: PrintThreeNumbers(0, false) ==> 0
Example: PrintThreeNumbers(12, false) ==> 12
Example: PrintThreeNumbers(123, false) ==> 123
Example: PrintThreeNumbers(0, true) ==> 000
Example: PrintThreeNumbers(12, true) ==> 012
Example: PrintThreeNumbers(123, true) ==> 123

```
void PrintThreeNumbers(int i, bool zeroFill)
{
        if (i < 10 && zeroFill)
                cout << "00";
        else if (i < 100 && zeroFill)
                cout << "0";
        cout << i;
}
```

FormatNumber - prints a number between 0 and 999,999 with commas
Hint: Use the PrintThreeNumbers function above
Example FormatNumber(999999) ==> 999,999
Example FormatNumber(0) ==> 0
Example FormatNumber(12423) ==> 12,423
Example FormatNumber(12000) ==> 12,000

```
void FormatNumber(int i)
{
	int firstPart = i / 1000;
	int secondPart = i % 1000;
	if (firstPart)
		cout << firstPart << ",";
	PrintThreeNumbers(secondPart, firstPart > 0);
}
```