

MIDTERM 2 CS111 Fall 2019

NAME _____ CUNYID _____

Instructions:

- 1) Make sure your name and CUNY ID are filled in.
- 2) For programming questions, in most cases, I have written the signature of a function for you, and you just have to fill in the rest of the function.
- 3) When asked for loop output:
 - a) for endless loops, write the first outputted character, then three dots, e.g. "2..."
 - b) If there is no output, write "No Output"
- 4) You can only use functions that are used in this test or ones that you write yourself. The only exception is that you can use the cout, rand, and time functions.

SECTION 1 – Miscellaneous Questions -5 Points Each

Q1) Is it good practice to use global variables. Why or why not?

No, Because it is very hard to follow program execution and discover which function modified the global variable.

Q2) Why will the function f1() below crash if called?

It will blow the call stack.

```
int callCount = 0;
void f1()
{
    callCount++;
    cout << callCount << endl;
    f1();
}
```

Q3) Is there anything wrong with the function below? If there are any mistakes, correct them.

Yes. It won't do anything to a and b in the calling program. The function signature should call by reference, Swap(int &a, int &b).

```
void Swap(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
```

Q4) What is the output of calling the statement below that calls Multiply3?

Q4) 0 (a * b * * 0 = 0)

```
cout << Multiply3(1, 2);
int Multiply3(int a=0, int b=0, int c =0)
{
    return(a*b*c);
}
```

Q5) What is the output of calling the statement below (f1(f2(f3()))); ?

Q5) 3 3 3

```
f1(f2(f3()));
int f3(int i = 3) { cout << i << " "; return(i); }
int f2(int i = 2) { cout << i << " "; return(i); }
int f1(int i = 1) { cout << i << " "; return(i); }
```

Q6) Is there anything wrong with the max statement below?

Q6) If you call max(-1, -3), max will return 0 an unexpected result.

```
int max(int i1, int i2, int i3 = 0)
{
    int temp = i1 > i2 ? i1 : i2;
    return(temp > i3 ? temp : i3);
}
```

SECTION 2 – What is the loop output? -5 Points Each**Instructions: Write the output for each loop below.****For endless loops, write the first outputted character then three dots, e.g. “2...”****If there is no output, write “No Output.”**

Q1)

```
int i = 0,j=0;
```

```
while (i * j <= 25)
```

```
{
```

```
    i++;
```

```
    if (i % 2 == 1)
```

```
        continue;
```

```
    if (i * j == 25)
```

```
        break;
```

```
    j++;
```

```
    cout << i << " ";
```

```
}
```

```
cout << endl;
```

Q1) 2 4 6 8

Q2)

```
for (char c = 'z'; c >= 'a'; c--)
```

```
{
```

```
    if (c <= 'w' || c >= 'c')
```

```
        continue;
```

```
    cout << c << " ";
```

```
}
```

```
cout << endl;
```

Q2) No Output

Q3)

```
int i = 0;
do
{
    i++;
    cout << i << endl;
} while (i / 7 != 0);
```

Q3) 1

Q4)

```
for (int i = 0; i < 3; i++)
{
    for (int j = i; j > 0; j--)
    {
        int output = i * j;
        if(output)
            cout << i * j << " ";
    }
}
```

Q4) 1 4 2

Q5)

```
bool a = true, b = true;
int i = 0;
while (a || b)
{
    i++;
    a = i % 2 == 0;
    b = i % 2 != 0;
    cout << i << " ";
}
```

Q5) 1...

Q6)

```
bool a = true, b = true;
int i = 1;
while (a || b)
{
    i++;
    a = i % 2 == 0;
    b = i % 3 == 0;
    cout << i << " ";
}
```

Q6) 2 3 4 5

Q7)

```
int sum = 0;

srand(time(0));

while (sum < 1000)
    sum += rand() % 2;

cout << sum << endl;
```

Q7) 1000

SECTION 3 – Fun with loops! -10 Points Each

Q1)

Replace the while loop below with a for loop

1) discover the output of the loop below

2) duplicate the output with a “for” loop

3) the only code within the “for” loop should be the statement `cout << i * j << " "`;e.g replace the "#####" with code so that its output is the same as the “while” loop
for (#####)`cout << i * j << " "`;

```
int i = 10;
int j = 0;
while (i >= j)
{
    if ((i % 2 == 0) || (j % 2 == 0))
    {
        i--;
        j++;
        continue;
    }
    cout << i * j << " ";
    i--;
    j++;
}
```

Q1) Answer for Q1). Complete the “for” loop below...

`// output is 9 21 25`

```
for(int i = 9, j = 1; i >= j; i -= 2, j += 2)
    cout << i * j << " "
```

SECTION 4 – Programming -15 Points each.**Helper functions that may assist you....**

```
double Absolute(double d) {return(d < 0 ? -d : d); }
```

```
int Sign(double d) {return(d < 0 ? -1 : 1); }
```

Q1) Write a power function that, given the following input, will provide the following output.
Note: decimal output can be in the form of a fraction or a decimal.

So, for example, Power(3,-2) → can be **1/9** or **0.111111**; Power(3,2) → → **9** Power(2,2) → **4**
More sample inputs and outputs...

| | Power | | | | |
|--------|-------|------|---|----|---|
| Number | -2 | -1 | 0 | 1 | 2 |
| 3 | 1/9 | 1/3 | 1 | 3 | 9 |
| 2 | 1/4 | 1/2 | 1 | 2 | 4 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| -1 | 1 | -1 | 1 | -1 | 1 |
| -2 | 1/4 | -1/2 | 1 | -2 | 4 |
| -3 | 1/9 | -1/3 | 1 | -3 | 9 |

```
// algorithm - Save sign and treat power as
```

```
// a positive number until returning result
```

```
double Power(int Number, int power)
```

```
{
```

```
    double result = 1;
```

```
    int sign = Sign(power);
```

```
    power = Absolute(power);
```

```
    while (power)
```

```
    {
```

```
        result *= Number;
```

```
        power--;
```

```
    }
```

```
    if (sign > 0)
```

```
        return(result);
```

```
    else
```

```
        return(1/result);
```

```
}
```

Q2)

AllPossibleThreeFactors - all possible three factors of a number

Note: a factor must be greater than 1 and less than the number itself. Also, many numbers will not have three factors.

AllPossibleThreeFactors(12) → 2 2 3; 2 3 2; 3 2 2


```

AllPossibleThreeFactors(2) → [no factors]
// algorithm - try every possibility of three
// variables i.e. triple for loop
void AllPossibleThreeFactors(int n)
{
    bool noResult = true;
    for (int i = 2; i < n; i++)
        for (int j = 2; j < n; j++)
            for (int k = 2; k < n; k++)
                if (i*j*k == n)
                {
                    noResult = false;
                    cout << i << " " <<
                        j << " " << k << " "
                        << endl;
                }
    if (noResult)
        cout << "No Result" << endl;
}

```

Q3) NthPrime and IsPrime – Returns the Nth prime number (for this function the first 10 prime numbers are “2 3 5 7 11 13 17 19 23 29”).

IsPrime – returns true/false depending on whether a number is prime or not. A prime number is defined as a whole number that can only be divided by 1 and itself evenly (without a remainder).

First, write the IsPrime helper function to determine if a specific number is prime or not. Then use IsPrime in the NthPrime function.

Example IsPrime(3) → true; IsPrime(4) → false

```

//Algorithm - try every possibility between
// 2 and n -1. If it divides evenly it is NOT
// a prime number, otherwise it is.

```

```

bool IsPrime(int n)
{
    for (int i = 2; i < n; i++)
        if (n%i == 0)
            return(false);
    return(true);
}

```

Example:

NthPrime(1) → 2; NthPrime(3) → 5

```
// Algorithm - start with 2 we try every
// succeeding number. If a number is prime
// we decrement count. When count is 0 we have
// found the N'th prime
int NthPrime(int count)
{
    int i = 2;
    while (count)
    {
        if (IsPrime(i))
        {
            count--;
            if (count == 0)
                return(i);
        }
        i++;
    }
}
```