## [FAST NU, LAHORE CAMPUS] [CS-101: INTRODUCTION TO COMPUTER SCIENCE]

**[MIDTERM 2] [TIME: 90 MIN] [TOTAL MARKS: 65]**

**[Fall 2011] [18th November, 2011]**

**Note:** a. Write clearly in the space provided

b. You are not allowed to ask questions during the exam. Make a reasonable assumption in case of any ambiguity

Good luck!

## Question 1 (3+3+4 points)

**Part a:** How many (‘\*’) will be printed. Circle the correct answer.

for (int i = 0; i <= 512; i\*=2)   
 cout<<’\*’;

1. 10
2. 9
3. 256
4. Infinity

**Part b:** How many time ‘#’ will be printed if x is 0 initially. Circle the correct answer.

while (x = 6)   
 {  
 printf("\*");   
 x++;   
 }

1. 6
2. 5
3. 7
4. Infinity

**Part c:** What will be the output of the program?  
int main()  
{  
 int T;  
 cout<<”Enter temperature in Centigrade”;  
 cin>>T; // The user entered 13  
 cout<< 9/5\*T+32 <<” Fahrenheit”<<endl;  
}

**ANSWER:**

## Question 2 (15 points)

This program wants to print all the primes between N and M, Identify its errors and write the corrected code in the box:   
  
int main()  
{  
 int N, M;  
  
 IsPrime = true;  
 for(int K = N; K<=M;K++)  
 {  
 for(int D = 2;D<=K;D++)  
 {  
 if(K%D = = 0)  
 {  
 IsPrime = false;  
 }  
 else  
 {  
 IsPrime = true;  
 }  
 }  
 if(IsPrime)  
 {  
 cout<<K<<” is Prime”<<endl;  
 }   
 }   
  
}

# Question 3 (5+5 points)

Floor of a floating number N is the nearest integer ≤ N. e.g. http://latex.codecogs.com/gif.latex?\lfloor%206.5%20\rfloor%20=%206 similarly http://latex.codecogs.com/gif.latex?\lfloor%206%20\rfloor%20=%206 and http://latex.codecogs.com/gif.latex?\lfloor%20-6.5%20\rfloor%20=%20-7.

**Part a:** Write the following function:

\_\_\_\_ floor(\_\_\_\_\_\_)  
{

}

**Part b:** Ceiling of a floating number N is the nearest integer ≥ N e.g. http://latex.codecogs.com/gif.latex?\lceil%206.5%20\rceil%20=%207 and http://latex.codecogs.com/gif.latex?\lceil%206%20\rceil%20=%206 and http://latex.codecogs.com/gif.latex?\lceil%20-6.5%20\rceil%20=%20-6  
Write the following function:

\_\_\_\_\_ ceil(\_\_\_\_\_\_ )  
{

}

**Question 4 (10 points)**

Given the following program:

bool MysteryFunction(int arr[], int size, int &y)

{

int j,temp;

bool done = false;

for (j=size-1;j>=0;--j)

{

if (y > arr[j] )

{

temp = arr[j];

arr[j] = y;

y = temp;

done = true;

}

}

return done;

}

void main()

{

int z = 18;

int MyFirstArr[] = {10,13,17,19,25};

bool MoreMystery = MysteryFunction(MyFirstArr,5,z);

int z1 = 18;

int MySecondArr[] = {25,29,35,40,60,65};

bool AnotherMystery = MysteryFunction(MySecondArr,6,z1);

}

Write the values of the following variables at the end of the program:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| z | z1 | MoreMystery | AnotherMystery | Contents of MyFirstArray | Contents of MySecondArray |
|  |  |  |  |  |  |

What does the MysteryFunction do in general? (1 line)

**Question 5 (10+10 points)**

The nutty professor constructed a program to do the following in the given order:

1. Take an array of numbers e.g., an array with {4,7,8,11,20}
2. Compute the sum of squares of odd elements of the array, e.g., for the above array the sum would be 7\*7+11\*11
3. Construct another array containing only the positive even numbered items in the array and fill the rest of the array with the number -1, e.g., for the above, an array of size 7 would be {4,8,20,-1,-1,-1,-1}

The program he wrote, unfortunately, got corrupted and now only a part of it remains. You have to complete the program when given his partial program, so that it achieves the above task and compiles properly without errors:

int main()

{

const int NUTTY\_ARRAY\_SIZE=5;

const int EVEN\_ARRAY\_SIZE = 10;

int NuttyArray[] = {4,7,8,11,20};

int EvenArray[EVEN\_ARRAY\_SIZE] = {0};

int SumOfSquares = -1;

computeSumOfSquresOfOddItems(NuttyArray,NUTTY\_ARRAY\_SIZE,

SumOfSquares);

int TotalEvenNumbers = ConstructEvenArray (NuttyArray, EvenArray, NUTTY\_ARRAY\_SIZE,EVEN\_ARRAY\_SIZE);

return 0;

}

SOLUTION