National University of Computer and Emerging Sciences

FAST School of Computing **Fall-2022** **Islamabad Campus**

CS-1002: Programming Fundamentals

Serial No:

Sessional Exam-II
Total Time: 1 Hour
Total Marks: 50

Thursday, 10th November, 2022

_____

Signature of Invigilator

Course Instructors

Dr. Mudassar Aslam, Mr. Jawad Hassan

_____	_____	_____	_____
Student Name	Roll No.	Course Section	Student Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Attempt all of them. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work. You may use last page for logic building rough work.
3. After asked to commence the exam, please verify that you have **ten (10)** different printed pages including this title page. There are a total of **2** questions.
4. Calculator or any other item sharing is strictly prohibited.
5. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.

	Q-1	Q-2	Total
Marks Obtained			
Total Marks	**36**	**14**	**50**

Question 1 [3 x 12 = 36 Marks]

Write output of the following program code in the given output column. In case of no output write **NO Output** with the reason (No marks without mentioning the reason). Don't miss dry run otherwise NO MARKS will be given even with correct output. You are allowed to trace code execution directly in the code column.

`float x = 10;`
`while (x < 100)`
`  {`
`  x *= 5;`
`  x -= 10;`
`  }`
`cout << x << endl;`
	Output
	
	190

Dry Run	

`int MAX = 70;`
`for (char ch = 65; ch <= MAX; ++ch) {`
`int i = 'A';`
`while (true)`
`{`
`if (i++ % 2 == 0)`
`  continue;`
`if (i > ch)`
`  break;`
`cout << ch << " ";`
`}`
`cout << endl;`
`}`
	Output
	
	B
	C
	D D
	E E
	F F F

Dry Run [ASCII of Z is 90]	

```
int x, y = 4;
for (x = 2; x < y ; x+=2)
y = y + 1 % x;
cout << y << endl;
x = y;
do {
cout << --x << endl;
x *= 4;
} while (x <= 10);
```

Output

```
6
5
```

Dry Run

```
int i, j, sum = 5;

for (i = 0; i<5; i++)
 if (i % 2)
    for (j = 0; j <= 3; sum += j++);
 else
    for (j = 3; j>0; sum += --j);
 cout << sum;
```

Output

```
26
```

Dry Run

```
float calc(int y, int x) {
    return (y + x + 9.0 / 2);
}
int main()
{
    float i = 7.5;
    int j = 4.5;
    cout << calc(i, j) << endl;
}
```

Output

```
15.5
```

Dry Run

```
int i = 12, counter = 5;
     while (i - 1)
     {
          if (i%3==0)
          ++counter;
          i-=1;
     }
     cout << counter;
```

Output

```
9
```

Dry Run

```
int choice = 5;
switch (choice) {
case 4:
       cout << "\nI am in case 4";
case 5:
       cout << "\nI am in case 5";
default:
       cout << "\nI am in Default";
case 6:
       cout << "\nI am in case 6";
       break;
case 7:
       cout << "\nI am in case 7";
}
```

Output:

```
I am in case 5
I am in Default
I am in case 6
```

```
int mystery(int x, int  n)
{
      int val;
      val = 1;
      if (n >= 0)
      {
            if (n % 3 > 1)
                  val = val * x;
            else
                  val = val * 2;
      }
      return val;
}
int main()
{
cout << "The mysterious value is: " << mystery(5, 2);
}
```

Output

```
The mysterious value is: 5
```

Dry Run

```
int main()
{
  int y = 2, x = 4, temp = 0;
  temp = y == 2 ? x < 1 ? x + y + 4 : x + y - 4 : x + 9;
  cout << temp;
}
```

Output:
2

Dry Run

```
y == 2 ? x < 1 ? x + y + 4 : x + y - 4 : x + 9
x < 1 ? x + y + 4 : x + y - 4
x + y - 4
2
```

```
void calc(float& , int&, int&);
int main()
{
	int val1 = 2, val2= 3;
	float res;
	calc(res, val1, val2);
	cout << val1 << " " << val2<<endl;
	cout << "Reult is : " << res ;

return 0;
}

void calc(float& r, int& para1, int& para2)
{
	para1 = 4 + para2 * 2;
	para2 +=  3 * para1;

	r = para1 + para2 / 2.0;
}
```

Output:

10 33
Result is : 26.5

Dry Run

```
int mystery(int value)
{
	static int count = 3;
	value += count;
	return value;
}
int main()
{
	int val1;
	for (int c = 0; c <= 10; c+=3)
	{
		val1 = mystery(c);
		cout << val1 << endl;
	}
	return 0;
}
```

Output:

3
6
9
12

Dry Run

```
void fun1(int&);
void fun2(int&);
int fun3(int);

int main()
{
	int val = 3;
	fun1(val);
	cout << val;
	return 0;
}

void fun1(int &p1)
{
	p1++;
	fun2(p1);
	p1++;
}
void fun2(int &p2)
{
	p2 = fun3(p2);
	p2++;
}
int fun3(int p3)
{
	p3 = p3 * 3;
	return p3;
}
```

Output:

14

Dry Run

Question 2 [8 + 6 = 14 Marks]

a) Write a piece of C++ program code (full program not needed) using nested for loops that prints following pattern. Manipulators are not allowed. [8 marks]

```
#####*#####
####*#*####
###*###*###
##*#####*##
#*#######*#
*#########*
```

```cpp
#include <iostream>
using namespace std;

int main()
{
    int end1 = 5;
    int end2 = -1;
    for(int i=1; i<=6; i++)
    {
        for(int hash=1; hash<=end1; hash++)
            cout<<"#";
        cout<<"*";

        for(int hash=1; hash<=end2; hash++)
            cout<<"#";
        end2+=2;
        cout<<"*";

        for(int hash=1; hash<=end1; hash++)
            cout<<"#";

        cout<<endl;
        end1--;
    }
    return 0;
}
```

b) Write a function ***charcmp()*** that takes two characters as parameters and compares them. If both characters are alphabets and similar (or equal) then change their cases, i.e., from lower case to upper or from upper case to lower and returns true. The function returns false if both characters are not similar (or equal) or are not alphabet. Write appropriate prototype and definition of the function. Also **write a main program** which calls this function

```
#include <iostream>

/*Write a function charcmp() that takes two characters
as parameters and compares them.
If both characters are alphabets and similar (or equal)
then change their cases, i.e., from lower case to upper
or from upper case to lower and returns true.
The function returns false if both characters are not similar
(or equal) or are not alphabet. Write appropriate prototype
and definition of the function. Also write a main program
which calls this function*/
using namespace std;

bool charcmp(char &c1, char &c2); [0.5 mark]

int main()
{
    char c1,c2;

    c1='a'; c2='A';
    cout<<c1<<"\t"<<c2<<"\tReturned:";
    cout<<charcmp(c1, c2)<<"\t"; [0.5 mark]
    cout<<c1<<"\t"<<c2<<endl;

    c1='A'; c2='a';
    cout<<c1<<"\t"<<c2<<"\tReturned:";
    cout<<charcmp(c1, c2)<<"\t";
    cout<<c1<<"\t"<<c2<<endl;

    c1='A'; c2='A';
    cout<<c1<<"\t"<<c2<<"\tReturned:";
    cout<<charcmp(c1, c2)<<"\t";
    cout<<c1<<"\t"<<c2<<endl;

    c1='a'; c2='C';
    cout<<c1<<"\t"<<c2<<"\tReturned:";
    cout<<charcmp(c1, c2)<<"\t";
    cout<<c1<<"\t"<<c2<<endl;
```

```
    c1='C'; c2='a';
    cout<<c1<<"\t"<<c2<<"\tReturned:";
    cout<<charcmp(c1, c2)<<"\t";
    cout<<c1<<"\t"<<c2<<endl;

    c1='D'; c2='A';
    cout<<c1<<"\t"<<c2<<"\tReturned:";
    cout<<charcmp(c1, c2)<<"\t";
    cout<<c1<<"\t"<<c2<<endl;

    c1='3'; c2='A';
    cout<<c1<<"\t"<<c2<<"\tReturned:";
    cout<<charcmp(c1, c2)<<"\t";
    cout<<c1<<"\t"<<c2<<endl;
}

bool charcmp(char &c1, char &c2) //[correct 1 mark]
{
    bool isAlpha = false;

    //check that both are alphabets [1 mark]
    isAlpha = (c1>='a' && c1<='z') || (c1>='A' && c1<='Z');
    isAlpha = isAlpha && (c2>='a' && c2<='z') || (c2>='A' && c2<='Z');
    if (isAlpha==false)
        return false;

    //Both are lower case and equal [0.5 mark]
    if ((c1==c2) && (c1>='a' && c1<='z'))
    {
        c1 = c2 = c1-32; //or get 32 by expression 'a' - 'A'
        return true;
    }
    //Both are upper case and equal [0.5 mark]
    else if ((c1==c2) && (c1>='A' && c1<='Z'))
    {
        c1 = c2 = c1+32; //or get 32 by expression 'a' - 'A'
        return true;
    }
    //c1 is lower, c2 is upper but both are similar [1 mark]
    else if ((c1>='a' && c1<='z') && (c2>='A' && c2<='Z'))
    {
        if ((c1-32) == c2)
        {
            c1 -= 32;
            c2 += 32;
            return true;
        }
```

```
    }
    //c2 is lower, c1 is upper but both are similar
    else if ((c2>='a' && c2<='z') && (c1>='A' && c1<='Z'))
    {
        if ((c2-32) == c1)
        {
            c1 += 32;
            c2 -= 32;
            return true;
        }
    }
    return false; [1 mark]

}
```

Correct prototype, call and definition structure – 2
Both are alphabets – 1
Both are equal and correct return – 1
Both are similar but different case - 2

ROUGH WORK