COL 100. Minor 1 Exam

Friday August 24, 2018

Name:
Entry Number:
Group:

Notes:

- Total number of questions: 5. Max Marks: 20
- All answers should be written on the question paper itself.
- The last two sheets in the question paper are meant for rough work. If you run out of space, you can answer questions in this rough space. However, please clearly mention in the answer space for the appropriate question that we should look at the rough space for its answer.
- We will collect the question paper (including the last two sheets meant for rough work). We will not be collecting back any other rough sheets.

1. **Fibonacci Series:** In this problem, we will write a program to output the numbers in a Fibonacci series. A Fibonacci series starts with numbers 0 and 1, and the number at the i^{th} position ($i \geq 3$) is the sum of the numbers at the positions $i-1$ and $i-2$. For example, first few elements of a Fibonacci series are given as: $0, 1, 1, 2, 3, 5, 8, 13, 21, \cdots$ and so on. Complete the following program to print the first n numbers of the Fibonacci series. [5 points]

```
#include <iostream>
#include "simpio.h";
using namespace std;

int main() {

  int n = getInteger("n?");


  int prev1 = 1, prev2 = 0, fib;
  cout << "First " << n << " numbers of Fibonacci series are :";
  for( int i=1; i <= n; i++)
  {
    if(i==1) fib = 0;
    else if(i==2) fib = 1;
    else {
      fib = prev1 + prev2;
      prev2 = prev1;
      prev1 = fib;
    }
    cout << " " << fib;
  }
  cout << endl;
  return 0;
}
```

2. **Sum Digits:** We would like to write a function $sumDigits(int\ n)$ which inputs a postive integer n and returns the sum of the digits of n. For example, if $n = 356$, your function should return the value $3 + 5 + 6 = 14$. Complete the code below to achieve the above functionality. You can make use of the % (remainder) operator where $a\%b$ returns the remainder of dividing a by b where a and b both are integers, e.g., $6\%4 = 2$. [5 points]

```
//Function sumDigits - to sum the digits of a number
int sumDigits(int n){

  int sum=0;


  while(n > 0) {
    sum += n%10;
    n = n/10;
  }

  return sum;
}
```

3. We would like to write a program which (a) repeatedly inputs a number n from the user until the entered number is a postive odd number (b) prints the following pattern for the given value of n. For example, if $n = 7$, following pattern is printed. You can use the `getInteger()` function available in `simpio.h` or use `cin` (it is your choice). [4 points]

```
*
***
*****
*******
*****
***
*
```

Note that the middle row has 7 stars in the above pattern.

```
#include <iostream>
#include "simpio.h"
using namespace std;

int main() {

   int n;
   while(true) {

     cin >>  n; or n = getInteger("n?");

     if( n > 0 && n %2 ==1) break;
   }
   for(int i=1; i<=n; i=i+2) {
     for(int j=0; j<i; j++)
       cout << "*";
     cout << endl;
   }
   for(int i=n-2; i>0; i=i-2) {
     for(int j=0; j<i; j++)
```

```
        cout << "*";
      cout << endl;
   }
   return 0;
}
```

4. What will be the output of the following program? [3 points]

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

int main() {
  int j=10;
  for (int i=0;i<j;j--) {
     int j=i++;
     cout<<j<<" ";
  }
  cout << endl;
}
```

You need to justify your answer by clearly stating which variable definition(s) does each usage of i and j bind (or refer) to. You should also justify the number of times the for loop will be excuted.

– Output of above program is: 0 1 2 3 4

Which variables bind to which declarations:
– Variable i is declared in the statement of the for loop. All uses of i in the body of the for loop bind (or refer) to this i (declared in the for loop statement).
The first variable j is declared in the first statement of the function body (let's call this $j1$). The use of $j--$ in the for loop statement refers to $j1$.
The second variable j is declared in the body of the for loop (let's call this $j2$). The use of j in `cout` refers to $j2$. Variable j declared inside loop body is assigned with value of variable i (i++ implements post-increment operator i.e., value is incremented after the assignment) and is printed by cout.

Number of times loop will be executed:
– Variable i is declared and initialized to 0 at loop entry.
–Variable j is initialized to 10 at the start of the function. The j used in the for-loop statement binds (or refers) to this j ($j1$).This is used

for checking loop exit condition and is decremented by 1 per iteration. –In every loop iteration, i increments by 1 and $j1$ decrements by 1. In other words, the distance between i and $j1$ reduces by 2. The values taken by i and $j1$ in each loop iteration are: $(0, 10)$, $(1, 9)$, $(2, 8)$, $(3, 7)$, $(4, 6)$. The loop exits at $(5, 5)$ because $i < j$ check returns $false$ at this point. Hence, the loop is executed 5 times.

5. What happens when the following code is executed? [3 points]

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

int foo(int n) {
  if(n > n/2 + 2*3) {
    cout<<"Statement 1 is executed. n = "<<n<<endl;
  } else if((n>=4 || n<=12) && n!=7) {
    cout<<"Statement 2 is executed. n = "<<n<<endl;
    n = n/2 + 2;
  } else if(n%2 != 0) {
    cout<<"Statement 3 is executed. n = "<<n<<endl;
    n = n + 1;
  } else {
    cout<<"Statement 4 is executed. n = "<<n<<endl;
  }
  return n;
}


int main()
{
  int n =11;
  n = foo(n);
  while(n >= 4)
  {
    n = foo(n);
  }
  return 0;
}
```

The output is :
Statement 2 is executed. n = 11
Statement 3 is executed. n = 7
Statement 2 is executed. n = 8
Statement 2 is executed. n = 6

Statement 2 is executed. n = 5
Statement 2 is executed. n = 4
Statement 2 is executed. n = 4
.
.

At n=4, the program will be stuck in infinite loop forever for the following reason. After n becomes 4, the while loop inside main function still calls function foo (as n $\geq$ 4) Inside function foo for n=4, the second branch will be taken. The value of n is updated to n/2+2, i.e. 4. The same value n=4 is returned by the function foo.