**LAB JOURNAL 5**

Name: Saad Ahmad

Enrollment #: 01-134222-130

Class/Section: BS-CS-3A

**Task 1 :**

Give answers to the following.

|  |  |
| --- | --- |
| 1. | Write the output of the following program.  #include<iostream>  usingnamespace std;  int mystery(int,int);  int main()  {  int x=3,y=2;  cout<<"Result = "<<mystery(x,y);  return 0;  }  int mystery(inta, intb)  {  if(b==1)  returna;  else  returna + mystery(a, b-1);  }  Output: 6 |
| 2. | Let J and K be integers and suppose Q(J, K) is recursively defined by :  Trace and Find Q(5, 3).  10 |
| 3. | Let ‘a’ and ‘b’ be integers and suppose Q(a, b) is recursively defined by :  Find Q(14,3).  5 |
| 5. | Identify the problem with following recursive function.  void recurse( int count )  {  cout<< count <<"\n";  recurse ( count + 1 );  }  No base case  if(count < 100) |
| 4. | Given the following function, write the output if the user enters ‘abcz’ as input.  void rev()  {  char c;  cin>>c;  if(c!='z'){  rev();  cout<<c;  }  }  Output:cba |

**Task 2 :**

Implement the following exercises.

**Exercise 1**

|  |
| --- |
| Ackermann’s function is defined recursively on non-negative integers as follows.  A(m,n) = n+1 if m == 0  A(m,n) = A(m-1, 1) if m != 0, n == 0  A(m,n) = A(m-1, A(m, n-1)) if m != 0, n != 0  Implement it as a recursive function Ackermann(M,N) which takes two positive integers as input and returns a positive integer as result. Once implemented test your program by evaluating Ackermann(2,2). |

**Code:**

#include<iostream>

using namespace std;

int Ackermann(int m, int n) {

if (m == 0) {

return n + 1;

}

else if (m != 0 && n == 0) {

return Ackermann(m - 1, 1);

}

else if (m != 0 && n != 0) {

return Ackermann(m - 1, Ackermann(m, n - 1));

}

}

int main()

{

cout << Ackermann(2, 2) << endl;

}

**Output :**

A screenshot of a computer

Description automatically generated

**Exercise 2**

|  |
| --- |
| Write a function sum(int a[], int size) to (recursively) compute the sum of the elements in an array.  **Example Run :**  int arr[]={1,2,3,4} ;  int result = sum(arr,4) ;  cout<<result<<endl ; //Should print 10 |

**Code:**

#include<iostream>

using namespace std;

int sum(int a[],int size) {

if (size > 1) {

a[size-2] = a[size-1] + a[size-2];

size--;

return sum(a, size);

}

else {

return a[0];

}

}

int main()

{

int arr[] = { 1,2,3,4 };

int result = sum(arr,. 4);

cout << result << endl; //Should print 10

}

**Output :**

**A screenshot of a computer

Description automatically generated**

**Exercise 3**

|  |
| --- |
| Write a recursive function to print integers from a given number N to 0. When called as print (10), the function should print : 10 9 8 7 6 5 4 3 2 1 0 |

#include<iostream>

using namespace std;

void print(int temp) {

if (temp >= 0) {

cout << temp << " ";

temp -= 1;

print(temp);

}

else {

return;

}

}

int main()

{

print(10);

}

**Output :**

**A screenshot of a computer

Description automatically generated**

**Exercise 4**

|  |
| --- |
| Binomial coefficients are normally computed using the following formula.  equation33331  Binomial coefficients can also be computed using the following recursive definition.  equation33347  Write a C++ program to compute binomial coefficients using the mentioned recursive definition. |

**Code :**

#include <iostream>

using namespace std;

int binomial(int n, int m) {

if (m == 0 || n == m) {

return 1;

}

else {

return binomial(n - 1, m) + binomial(n - 1, m - 1);

}

}

int main() {

cout << binomial(5, 4);

}

**Output:**

**A screenshot of a computer

Description automatically generated**

**Implement the given exercises and get them checked by your instructor. If you are unable to complete the tasks in the lab session, deposit this journal alongwith your programs (printed or handwritten) before the start of the next lab session.**

|  |  |  |
| --- | --- | --- |
| **S No.** | **Exercise** | **Checked By:** |
| 1. | Exercise 1 |  |
| 2. | Exercise 2 |  |
| 3. | Exercise 3 |  |
| 3. | Exercise 4 |  |