National University of Computer and Emerging Sciences

Lab Manual

for

Programming Fundamental Lab

Course Instructor	Prof. Aftab Alam
Lab Instructor(s)	Ms. Huda Mr. Shahzaib
Section	BSCS-1N
Semester	Fall 2021

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objective

- Static Arrays
- Reading data from files
- Recursion

Problems

Practice Question

1. What is the output of the following program?

```
int main()
{
  char Myarray1[5] = {'h', 'l', 'o','o','l' };
  char Myarray2[5] = {'e', 'l', 'w', 'r', 'd'};
  char Myarray3[10];

  for (int i = 0; i< 10; i= i+2)
  {
     Myarray3[i] = Myarray1[i/2];
     Myarray3[i+1] = Myarray2[i/2];
  }
  for (int i = 0; i< 10; i= i+1)
     cout<<Myarray3[i];

  return 0;
}
```

2. What changes are needed in the above program to print a space between two words printed.

3. What is the output of following program?

```
int main()
{
  int Myarray1[10] = {-1 ,1, 2, 3, 4, 5, 6, 7, 8, 9};

  for(int i = 1; i< 5; i= i+1)
  {
     int k = i*2;
      cout<<Myarray1[i]<<" "<<Myarray1[k]<<" "<<Myarray1[k+1]<<endl;
  }


  return 0;
}
```

Practice Question

We know that functions are called on stack Recursion is a method of calling a function within the same functions so recursions uses stacks although recursion is also used for doing repetitions just like loops but these are more readables and demands a stopping mechanism. In this question, we will do an example.
Below is written a function in which we are printing an array using recursive calls:

```
#include<iostream>
using namespace std;
void PrintArray(int arr[], int n){

 if(n>0)
  PrintArray(arr,n-1);
 cout<<arr[n]<<endl;
}

int main(){
 int arr[]={1,2,3,4};
 PrintArray(arr,3);
 return 0;
}
```

Output: 1,2,3,4

Problem Statements

Question :1

One of the local banks is gearing up for a big advertising campaign and would like to see how long its customers are waiting for service by teller/service provider at drive-up window. Several employees have been asked to keep accurate records for the 24-hour drive-up service. The collected information, which is read from a file, consists of ID number of the teller; the time the customer arrived in hours and minutes (arrival time); the time the customer actually was served in hours and minutes (service time). There is one record in the file for each customer. A teller can provide services to many customers. Write a program that does the following:

a) Reads in the wait data from a file name ***Bank.txt***.
b) Computes the **Waiting Time** for each customer, in minutes. Waiting time for each customer is calculated by subtracting the service time by the arrival time.
c) Compute the **Average Service Delay Time** for each teller. Average service delay time is calculated for each teller by summing the waiting time of all customers served by that teller and dividing that sum by the total number of customers served by the teller.
d) Calculate **Average Waiting Time** of all customers. Average waiting time is the sum of all waiting time divided by the number of customers.
e) Prints a summary showing the Average Waiting Time and Average Service Delay.

Input
A file containing teller ID, arrival time, and service time. The times are broken up into integer hours and minutes according to a 24-hour clock.

Output

The program should print on the screen, the Mean Waiting Time and Average Service delay time for each teller.

In sample input file, first column represents the teller's Id, second and third columns represent the arrival time in hours and minutes respectively. Fourth and fifth columns represent the service time in hours and minutes respectively.

Sample Input File Bank.txt

1	*2*	*30*	*4*	*30*
2	*2*	*0*	*3*	*30*
2	*3*	*30*	*4*	*0*
1	*5*	*0*	*6*	*0*
3	*5*	*0*	*5*	*30*
3	*5*	*30*	*6*	*0*
4	*1*	*0*	*2*	*0*

Sample Output

Average Waiting Time:				*60*
Average Service Delay:				
1	*90*			
2	*60*			
3	*30*			
4	*60*			

Problem Statement

Question : 2

Create a C++ program by using a recursive function which takes n and an array as input and will return an array containing numbers from 1 to n such that each index will be having sum of 1 to till that index(inclusive).

Let's say the user enter 4 along with an an input array of size 4.The output will look like:

1, 3, 6, 10.