PRINCIPLES OF PROGRAMMING LANGUAGES LAB

ETCS-458

Faculty name: Mrs. Ruchi Goel

Student name: Syeda Reeha Quasar

Roll No.: 14114802719

Semester: 8th

Group: 8C7

Maharaja Agrasen Institute of Technology, PSP Area,

Sector – 22, Rohini, New Delhi – 110085

Index

9.	Write a program to display 10 questions at random out of exp.8-50 questions (do not display the answer of these questions to the user now)				
10.	Implement producer-consumer problem using threads				
Beyond the Syllabus Questions					
1.	Write a Program where it may or may not print counter value in sequence and every time we run it, it produces a different result based on CPU availability to a thread.				
2.	There are 200 questions on a 3 hr examination. Among these questions are 50 mathematics problems. It is suggested that twice as much time be spent on each maths problem as for each other question. WAP which calculates how many minutes should be spent on mathematics problems.				
3.	Two polynomials are entered by the user in the form of : ax2 + bx + c where the powers of x can be any integer value and a,b& c are constants. Now WAP in C and JAVA which calculates the sum, product and difference of the two polynomials.				
4.	The hexadecimal digits are the ordinary, base-10 digits '0' through '9' plus the letters 'A' through 'F'. In the hexadecimal system, these digits represent the values 0 through 15, respectively. Write a function in JAVA and C named hexValue that uses a switch statement to find the hexadecimal value of a given character. The character is a parameter to the function, and its hexadecimal value is the return value of the function. You should count lower case letters 'a' through 'f' as having the same value as the corresponding upper case letters. If				

	the parameter is not one of the legal hexadecimal digits, return -1 as the value of the function.				
5.	A coffee shop blends 2 kinds of coffee, putting in 2 parts of a 33p. a gm. grade to 1 part of a 24p. a gm. If the mixture is changed to 1 part of the 33p. a gm. to 2 parts of the filess expensive grade .WAP which calculates that how much will the shop save in blending 100 gms				
6.	In June a baseball team that played 60 games had won 30% of its game played. After a phenomenal winning streak this team raised its average to 50% . WAP which calculates how many games must the team have won in a row to attain this average.				
7.	A company contracts to paint 3 houses. Mr. Brown can paint a house in 6 days while Mr. Black would take 8 days and Mr. Blue 12 days. After 8 days Mr. Brown goes on vacation and Mr. Black begins to work for a period of 6 days. WAP which calculates how days will it take Mr. Blue to complete the contract				
8.	2 hours after a freight train leaves Delhi a passenger train leaves the same station traveling in the same direction at an average speed of 16 km/hr. After traveling 4 hrs the passenger train overtakes the freight train. WAP which calculates the average speed of the freight train.				

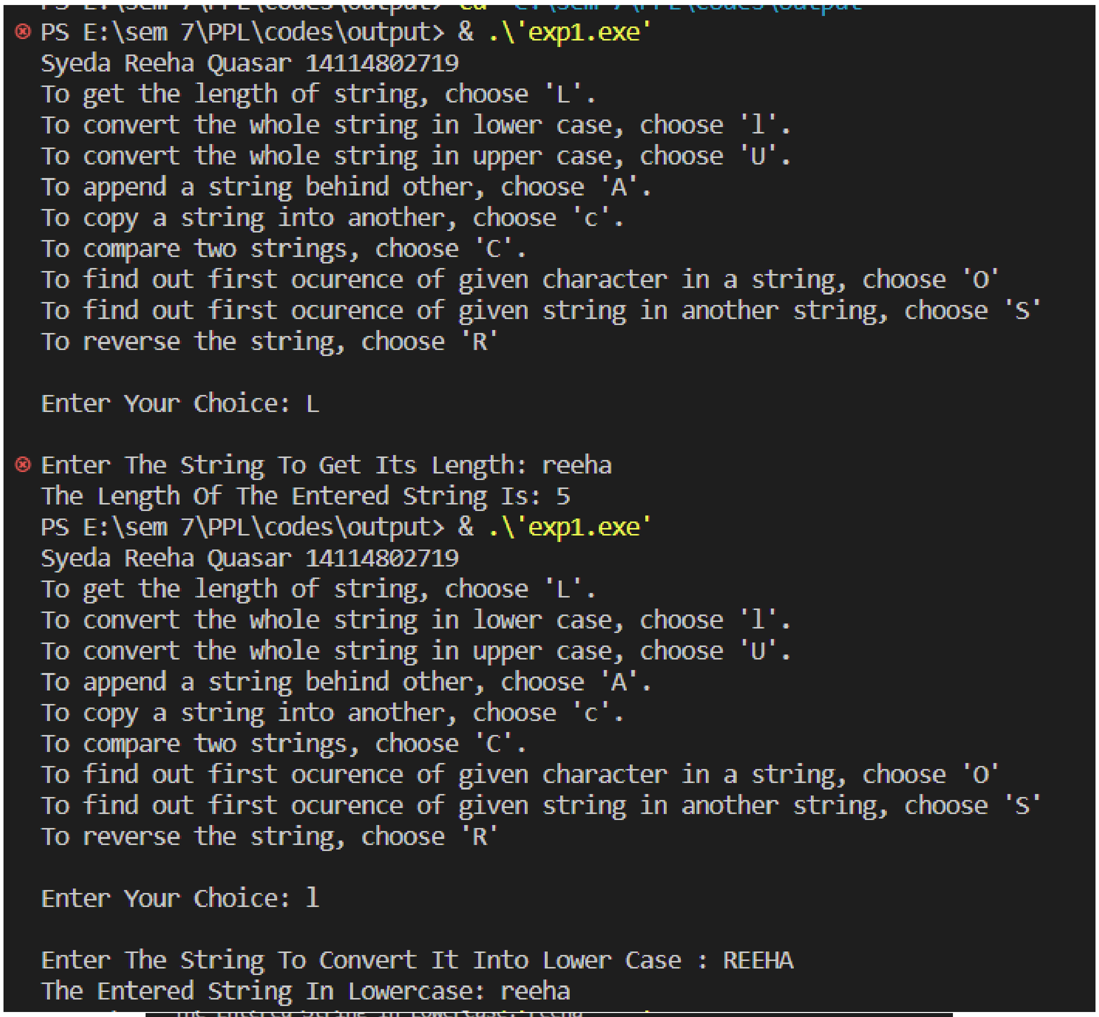
Experiment 1

Aim : Implement all major functions of string.h in single C program using switch case toselect specific function from user choice (like strlen, strcat, strcpy, strcmp, strrev)

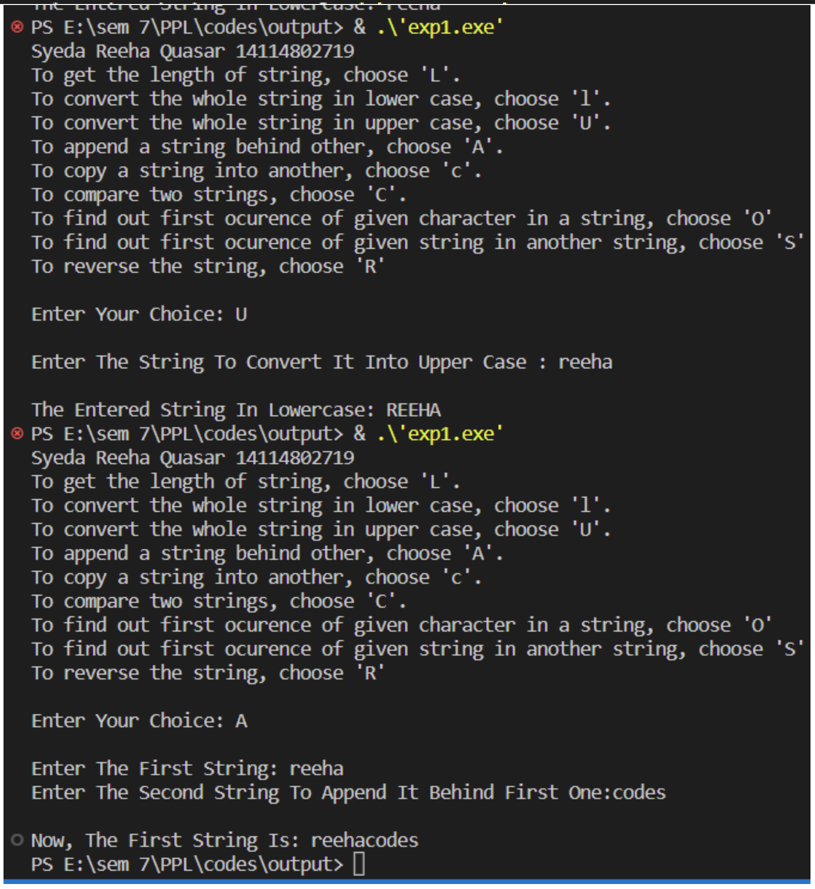
Code :

```
#include <stdio.h>
#include <string.h>

void main()
{
    char str[50];
    char temp[20];
    char choice, ch;
    // printf("For ")
    printf("Syeda Reeha Quasar 14114802719 \n");
    puts("To get the length of string, choose 'L'.");              // strlen();//
    puts("To convert the whole string in lower case, choose 'l'."); // strlwr();
    puts("To convert the whole string in upper case, choose 'U'."); // strupr();
    puts("To append a string behind other, choose 'A'.");           // strcat();//
    puts("To copy a string into another, choose 'c'.");             // strcpy();//
    puts("To compare two strings, choose 'C'.");                    // strcmp();//
    puts("To find out first ocurence of given character in a string, choose 'O'");
    // strchr();
    puts("To find out first ocurence of given string in another string, choose 'S'");
    // strstr();
    puts("To reverse the string, choose 'R'"); // strrev();//
    printf("\nEnter Your Choice: ");
    scanf("%c", &choice);
    switch (choice)
    {
    case 'L':
        printf("\nEnter The String To Get Its Length: ");
        scanf("%s", str);
        printf("The Length Of The Entered String Is: %d", strlen(str));
        break;
    case 'l':
        printf("\nEnter The String To Convert It Into Lower Case : ");
        scanf("%s", str);
        printf("The Entered String In Lowercase: %s", strlwr(str));
        break;
    case 'U':
        printf("\nEnter The String To Convert It Into Upper Case : ");
        scanf("%s", str);
        printf("\nThe Entered String In Lowercase: %s", strupr(str));
        break;
    case 'A':
        printf("\nEnter The First String: ");
        scanf("%s", str);
        printf("Enter The Second String To Append It Behind First One:");
        scanf("%s", temp);
        strcat(str, temp);
        printf("\nNow, The First String Is: %s", str);
        break;
    case 'c':
        printf("\nEnter The First String: ");
        scanf("%s", str);
        printf("Enter The Second String: ");
        scanf("%s", temp);
        strcpy(str, temp);
        printf("\nNow, The First String Is: %s", str);
        printf("\nAnd, The Second String Is: %s", temp);
        break;
    case 'C':
        printf("\nEnter The First String: ");
        scanf("%s", str);
        printf("Enter The Second String: ");
        scanf("%s", temp);
        if (strcmp(str, temp) == 0)
            printf("\nBoth Strings Are Similar.");
        else
            printf("\nBoth Strings Are Different.");
        break;
    case 'O':
        printf("\nEnter The String: ");
        scanf("%s", str);
        printf("Enter The Character To Be Searched: ");
        scanf("%c", &ch);
        printf("\nThe First Occurence of Character Is At: %s", strchr(str, ch));
        break;
    case 'S':
        printf("\nEnter The String: ");
        scanf("%s", str);
        printf("Enter The String To Be Searched: ");
        scanf("%s", temp);
        printf("\nThe First Occurence of Character Is At: %s", strstr(str, temp));
        break;
    case 'R':
        printf("\nEnter The String To Get Its Reverse: ");
        scanf("%s", str);
        printf("\nThe Reverse Of The Entered String Is: %s", strrev(str));
        break;
    default:
        printf("\nYou Entered A Wrong Choise.");
        break;
    }
}
```

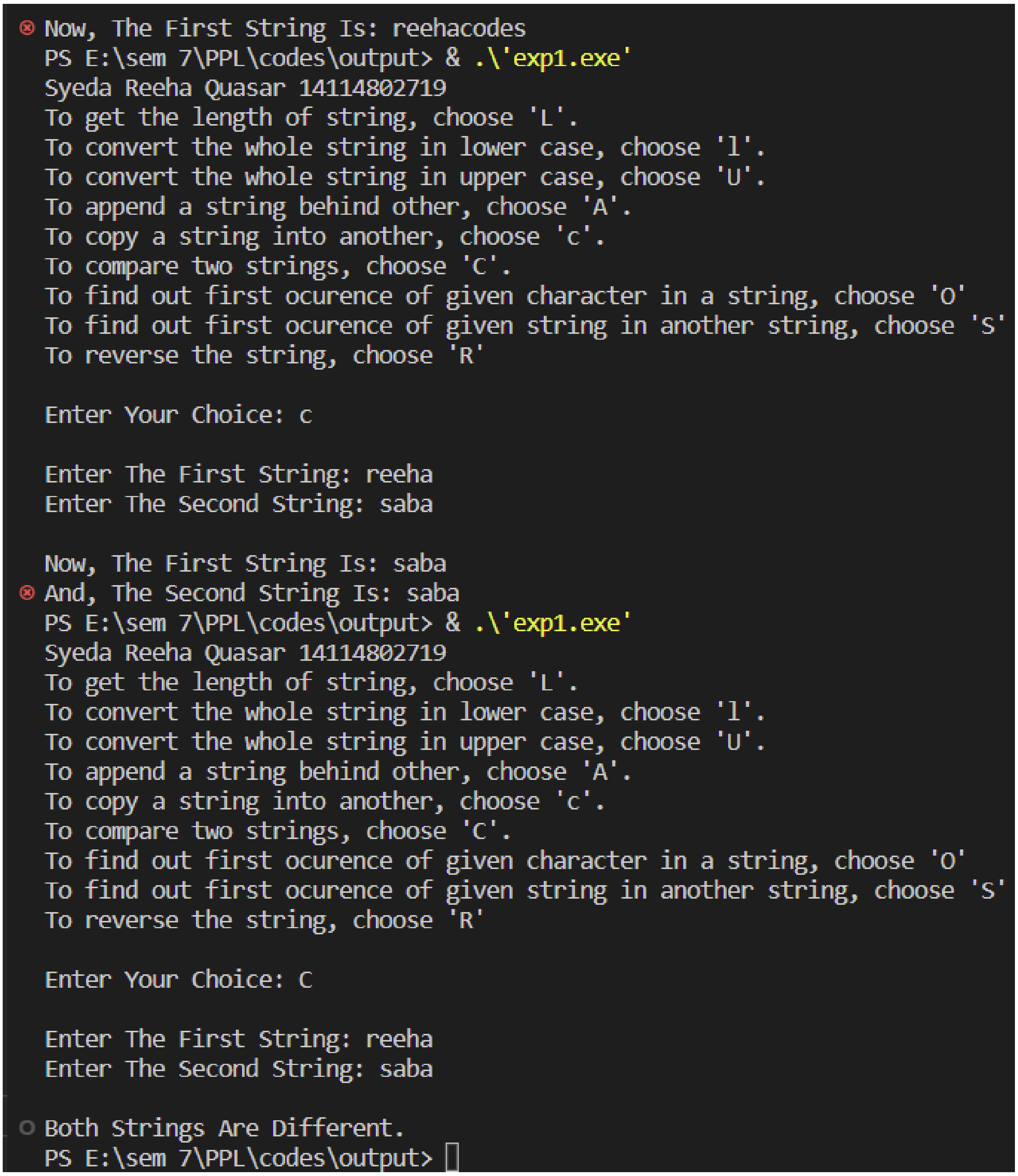
Output :



```
PS E:\sem 7\PPL\codes\output> & .\'exp1.exe'
Syeda Reeha Quasar 14114802719
To get the length of string, choose 'L'.
To convert the whole string in lower case, choose 'l'.
To convert the whole string in upper case, choose 'U'.
To append a string behind other, choose 'A'.
To copy a string into another, choose 'c'.
To compare two strings, choose 'C'.
To find out first ocurence of given character in a string, choose 'O'
To find out first ocurence of given string in another string, choose 'S'
To reverse the string, choose 'R'

Enter Your Choice: L

Enter The String To Get Its Length: reeha
The Length Of The Entered String Is: 5
PS E:\sem 7\PPL\codes\output> & .\'exp1.exe'
Syeda Reeha Quasar 14114802719
To get the length of string, choose 'L'.
To convert the whole string in lower case, choose 'l'.
To convert the whole string in upper case, choose 'U'.
To append a string behind other, choose 'A'.
To copy a string into another, choose 'c'.
To compare two strings, choose 'C'.
To find out first ocurence of given character in a string, choose 'O'
To find out first ocurence of given string in another string, choose 'S'
To reverse the string, choose 'R'

Enter Your Choice: l

Enter The String To Convert It Into Lower Case : REEHA
The Entered String In Lowercase: reeha
```



```
PS E:\sem 7\PPL\codes\output> & .\'exp1.exe'
Syeda Reeha Quasar 14114802719
To get the length of string, choose 'L'.
To convert the whole string in lower case, choose 'l'.
To convert the whole string in upper case, choose 'U'.
To append a string behind other, choose 'A'.
To copy a string into another, choose 'c'.
To compare two strings, choose 'C'.
To find out first ocurence of given character in a string, choose 'O'
To find out first ocurence of given string in another string, choose 'S'
To reverse the string, choose 'R'

Enter Your Choice: U

Enter The String To Convert It Into Upper Case : reeha

The Entered String In Lowercase: REEHA
PS E:\sem 7\PPL\codes\output> & .\'exp1.exe'
Syeda Reeha Quasar 14114802719
To get the length of string, choose 'L'.
To convert the whole string in lower case, choose 'l'.
To convert the whole string in upper case, choose 'U'.
To append a string behind other, choose 'A'.
To copy a string into another, choose 'c'.
To compare two strings, choose 'C'.
To find out first ocurence of given character in a string, choose 'O'
To find out first ocurence of given string in another string, choose 'S'
To reverse the string, choose 'R'

Enter Your Choice: A

Enter The First String: reeha
Enter The Second String To Append It Behind First One:codes

Now, The First String Is: reehacodes
PS E:\sem 7\PPL\codes\output> 
```



```
⊗ Now, The First String Is: reehacodes
  PS E:\sem 7\PPL\codes\output> & .\'exp1.exe'
  Syeda Reeha Quasar 14114802719
  To get the length of string, choose 'L'.
  To convert the whole string in lower case, choose 'l'.
  To convert the whole string in upper case, choose 'U'.
  To append a string behind other, choose 'A'.
  To copy a string into another, choose 'c'.
  To compare two strings, choose 'C'.
  To find out first ocurence of given character in a string, choose 'O'
  To find out first ocurence of given string in another string, choose 'S'
  To reverse the string, choose 'R'

  Enter Your Choice: c

  Enter The First String: reeha
  Enter The Second String: saba

  Now, The First String Is: saba
⊗ And, The Second String Is: saba
  PS E:\sem 7\PPL\codes\output> & .\'exp1.exe'
  Syeda Reeha Quasar 14114802719
  To get the length of string, choose 'L'.
  To convert the whole string in lower case, choose 'l'.
  To convert the whole string in upper case, choose 'U'.
  To append a string behind other, choose 'A'.
  To copy a string into another, choose 'c'.
  To compare two strings, choose 'C'.
  To find out first ocurence of given character in a string, choose 'O'
  To find out first ocurence of given string in another string, choose 'S'
  To reverse the string, choose 'R'

  Enter Your Choice: C

  Enter The First String: reeha
  Enter The Second String: saba

○ Both Strings Are Different.
  PS E:\sem 7\PPL\codes\output> ▯
```

Viva Questions

Q1. What is the use of string.h header while and where is this file stored.
In C programming language, string.h is a header that defines one variable type, one macro,and various functions for manipulating arrays of characters. All C inbuilt functions are declared in string.h header file.

Q2. How can we create a header file?
To make a header file, we have to create one file with a name, and the extension should be(*.h). In that function, there will be no main() function. In that file, we can put some variables, some functions, etc.

Q3. Write the function to find the length of a string.
The strlen() function in C is used to calculate the length of a string.

Q4. Write the function to concatenate two strings.
The best way to concatenate two strings in C programming is by using the strcat() function.

Q5. Explain the use of the header file.
Header files are simply files in which you can declare your own functions that you can use inyour main program or these can be used while writing large C programs. The header file implementation brings lots of readability to our program and it becomes easy to understand. If it is the way to write our code systematically and the header file brings abstraction,
standardization, and loose coupling between our main function file(.c) and other (.c) fileswhich we are using.

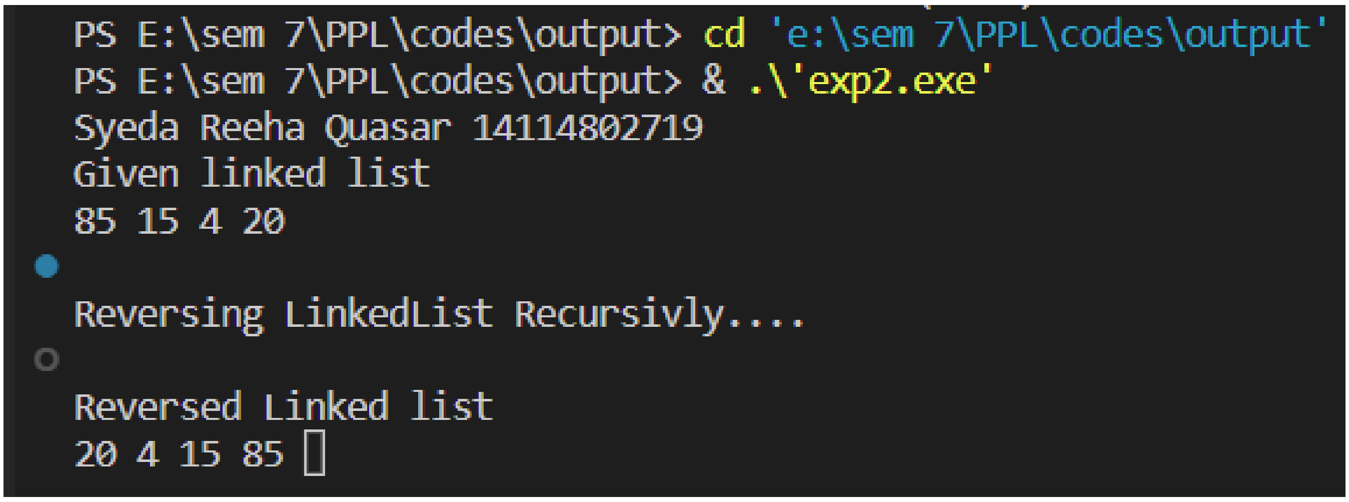
Experiment 2

Aim : Write a program (WAP) in C to reverse a linked list iterative and recursive.

Code :

```
// Iterative C program to reverse a linked list
#include <stdio.h>
#include <stdlib.h>
/* Link list node */
struct Node
{
   int data;
   struct Node *next;
};
/* Function to reverse the linked list */
void recursiveReverse(struct Node *head, struct Node **headRef)
{
   struct Node *first;
   struct Node *rest;
   // empty list base case
   if (head == NULL)
      return;
   first = head;      // suppose first = {1, 2, 3}
   rest = first->next; // rest = {2, 3}
   // base case: List has only one node
   if (rest == NULL)
   {
      // fix the head pointer here
      *headRef = first;
      return;
   }
   // Recursively reverse the smaller {2, 3} case
   // after: rest = {3, 2}
   recursiveReverse(rest, headRef);
   // put the first elem on the end of the list
   rest->next = first;
   first->next = NULL; // (tricky step -- make a drawing)
}
void reverse(struct Node **head_ref)
{
   struct Node *prev = NULL;
   struct Node *current = *head_ref;
   struct Node *next = NULL;
   printf("\n\nReversing Linked List Ittereatively...\n");
   while (current != NULL)
   {
      // Store next
      next = current->next;
      // Reverse current node's pointer
      current->next = prev;
      // Move pointers one position ahead.
      prev = current;
      current = next;
   }
   *head_ref = prev;
}
/* Function to push a node */
void push(struct Node **head_ref, int new_data)
{
    struct Node *new_node =
        (struct Node *)malloc(sizeof(struct Node));
    new_node->data = new_data;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}
/* Function to print linked list */
void printList(struct Node *head)
{
    struct Node *temp = head;
    while (temp != NULL)
    {
        printf("%d ", temp->data);
        temp = temp->next;
    }
}
int main()
{
    printf("Syeda Reeha Quasar 14114802719 \n");
    /* Start with the empty list */
    struct Node *head = NULL;
    push(&head, 20);
    push(&head, 4);
    push(&head, 15);
    push(&head, 85);
    printf("Given linked list\n");
    printList(head);
    // reverse(&head);
    printf("\n\nReversing LinkedList Recursivly....\n");
    recursiveReverse(head, &head);
    printf("\nReversed Linked list \n");
    printList(head);
    getchar();
}
```

Output :



```
PS E:\sem 7\PPL\codes\output> cd 'e:\sem 7\PPL\codes\output'
PS E:\sem 7\PPL\codes\output> & .\'exp2.exe'
Syeda Reeha Quasar 14114802719
Given linked list
85 15 4 20

Reversing LinkedList Recursivly....

Reversed Linked list
20 4 15 85
```

Viva Questions:

Q1. What is the difference between iterative and recursive function call?
Recursion and iteration are both different ways to execute a set of instructions repeatedly. Themain difference between these two is that in recursion, we use function calls to execute the statements repeatedly inside the function body, while in iteration, we use loops like “for” and “while” to do the same. Iteration is faster and more space-efficient than recursion, whereas it's easier to code a recursive approach for a given problem.

Q2. What are formal parameters in functions?
Formal parameters are the variables defined by the function that receives values when thefunction is called.

Q3. How is the structure node declared?
The general syntax for a struct declaration in C is:
struct tag_name {type member1; type member2;
/* declare as many members as desired, but the entire structure size must be known to thecompiler. */
};

Q4. Define a link list.
A linked list is a sequence of data structures, which are connected together via links. LinkedList is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array.

Q5. Why do we need to store the address of the starting node of a link list for reversinga list?
To reverse a LinkedList iteratively, we need to store the references of the next and previous elements, so that they don't get lost when we swap the memory address pointers to the nextelement in the LinkedList.

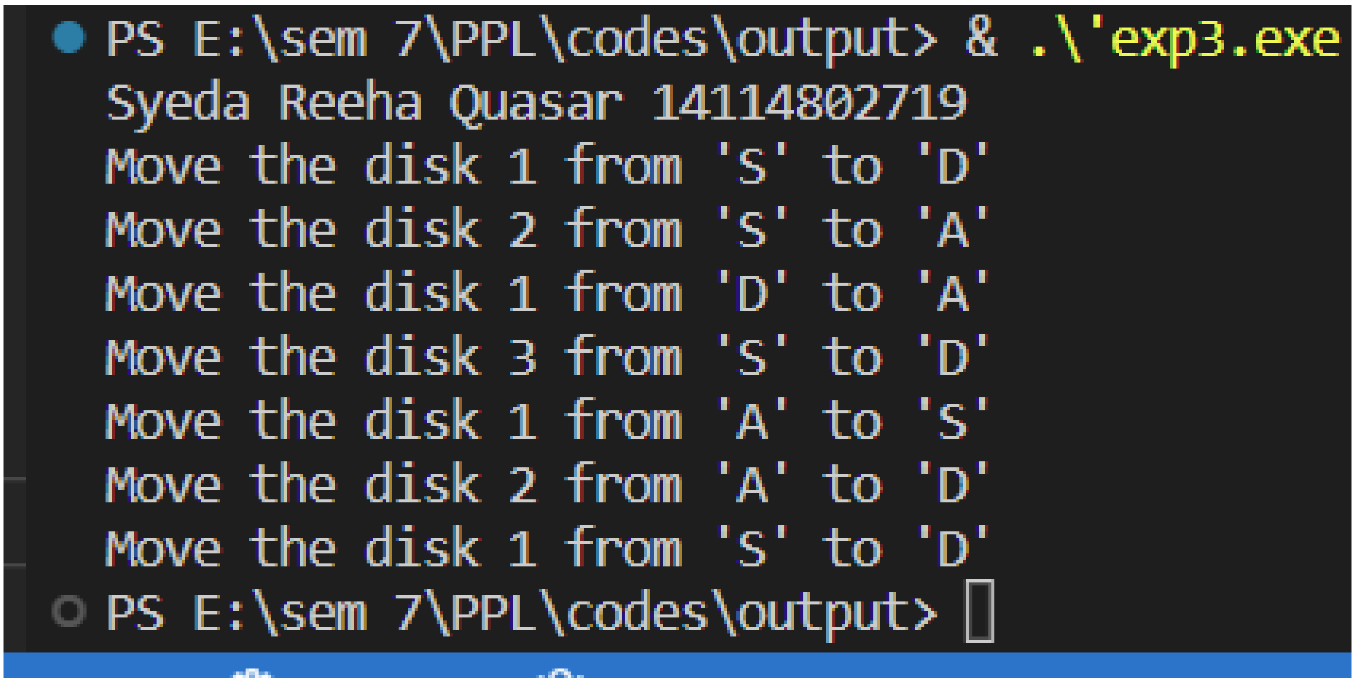
Experiment 3

Aim : WAP in C to implement iterative Towers of Hanoi.

Code :

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <limits.h>
// A structure to represent a stack
struct Stack
{
    unsigned capacity;
    int top;
    int *array;
};
// function to create a stack of given capacity.
struct Stack *createStack(unsigned capacity)
{
    struct Stack *stack =
        (struct Stack *)malloc(sizeof(struct Stack));
    stack->capacity = capacity;
    stack->top = -1;
    stack->array =
        (int *)malloc(stack->capacity * sizeof(int));
    return stack;
}
// Stack is full when top is equal to the last index
int isFull(struct Stack *stack)
{
    return (stack->top == stack->capacity - 1);
}
// Stack is empty when top is equal to -1
int isEmpty(struct Stack *stack)
{
    return (stack->top == -1);
}
// Function to add an item to stack. It increases
// top by 1
void push(struct Stack *stack, int item)
{
    if (isFull(stack))
        return;
    stack->array[++stack->top] = item;
}
// Function to remove an item from stack. It
// decreases top by 1
int pop(struct Stack *stack)
{
    if (isEmpty(stack))
        return INT_MIN;
    return stack->array[stack->top--];
}
// Function to show the movement of disks
void moveDisk(char fromPeg, char toPeg, int disk)
{
    printf("Move the disk %d from \'%c\' to \'%c\'\n",
           disk, fromPeg, toPeg);
}
// Function to implement legal movement between
// two poles
void moveDisksBetweenTwoPoles(struct Stack *src,
                              struct Stack *dest, char s, char d)
{
    int pole1TopDisk = pop(src);
    int pole2TopDisk = pop(dest);
    // When pole 1 is empty
    if (pole1TopDisk == INT_MIN)
    {
        push(src, pole2TopDisk);
        moveDisk(d, s, pole2TopDisk);
    }
    // When pole2 pole is empty
    else if (pole2TopDisk == INT_MIN)
    {
        push(dest, pole1TopDisk);
        moveDisk(s, d, pole1TopDisk);
    }
    // When top disk of pole1 > top disk of pole2
    else if (pole1TopDisk > pole2TopDisk)
    {
        push(src, pole1TopDisk);
        push(src, pole2TopDisk);
        moveDisk(d, s, pole2TopDisk);
    }
    // When top disk of pole1 < top disk of pole2
    else
    {
        push(dest, pole2TopDisk);
        push(dest, pole1TopDisk);
        moveDisk(s, d, pole1TopDisk);
    }
}
// Function to implement TOH puzzle
void tohIterative(int num_of_disks, struct Stack *src, struct Stack *aux,
                  struct Stack *dest)
{
    int i, total_num_of_moves;
    char s = 'S', d = 'D', a = 'A';
    // If number of disks is even, then interchange
    // destination pole and auxiliary pole
    if (num_of_disks % 2 == 0)
    {
        char temp = d;
        d = a;
        a = temp;
    }
    total_num_of_moves = pow(2, num_of_disks) - 1;
    // Larger disks will be pushed first
    for (i = num_of_disks; i >= 1; i--)
        push(src, i);
    for (i = 1; i <= total_num_of_moves; i++)
    {
        if (i % 3 == 1)
            moveDisksBetweenTwoPoles(src, dest, s, d);
        else if (i % 3 == 2)
            moveDisksBetweenTwoPoles(src, aux, s, a);
        else if (i % 3 == 0)
            moveDisksBetweenTwoPoles(aux, dest, a, d);
    }
}
// Driver Program
int main()
{
    printf("Syeda Reeha Quasar 14114802719 \n");
    // Input: number of disks
    unsigned num_of_disks = 3;
    struct Stack *src, *dest, *aux;
    // Create three stacks of size 'num_of_disks'
    // to hold the disks
    src = createStack(num_of_disks);
    aux = createStack(num_of_disks);
    dest = createStack(num_of_disks);
    tohIterative(num_of_disks, src, aux, dest);
    return 0;
}
```

Output :



```
PS E:\sem 7\PPL\codes\output> & .\'exp3.exe
Syeda Reeha Quasar 14114802719
Move the disk 1 from 'S' to 'D'
Move the disk 2 from 'S' to 'A'
Move the disk 1 from 'D' to 'A'
Move the disk 3 from 'S' to 'D'
Move the disk 1 from 'A' to 'S'
Move the disk 2 from 'A' to 'D'
Move the disk 1 from 'S' to 'D'
PS E:\sem 7\PPL\codes\output>
```

Viva Questions:

Q1. What is the tower of Hanoi problem?
Tower of Hanoi is a mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move the entire stack to another rod, obeying the followingsimple rules:

- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it ontop of another stack i.e. a disk can only be moved if it is the uppermost disk on a
stack.
- No disk may be placed on top of a smaller disk.

Q2. What are the various ways is stack created?
There are two ways to implement a stack: Using array. Using linked list.

Q3. List the models of computation of language.
Models of computation can be classified into three categories: sequential models, functionalmodels, and concurrent models.

Q4. What are objectives of principle of programming language?
Objectives are: To introduce several different paradigms of programming. To gain experiencewith these paradigms by using example programming languages. To understand concepts of syntax, translation, abstraction, and implementation.

Q5. What are the Paradigms of Programming?
Major Programming Paradigms

- Imperative.
- Logical.
- Functional.
- Object-Orient

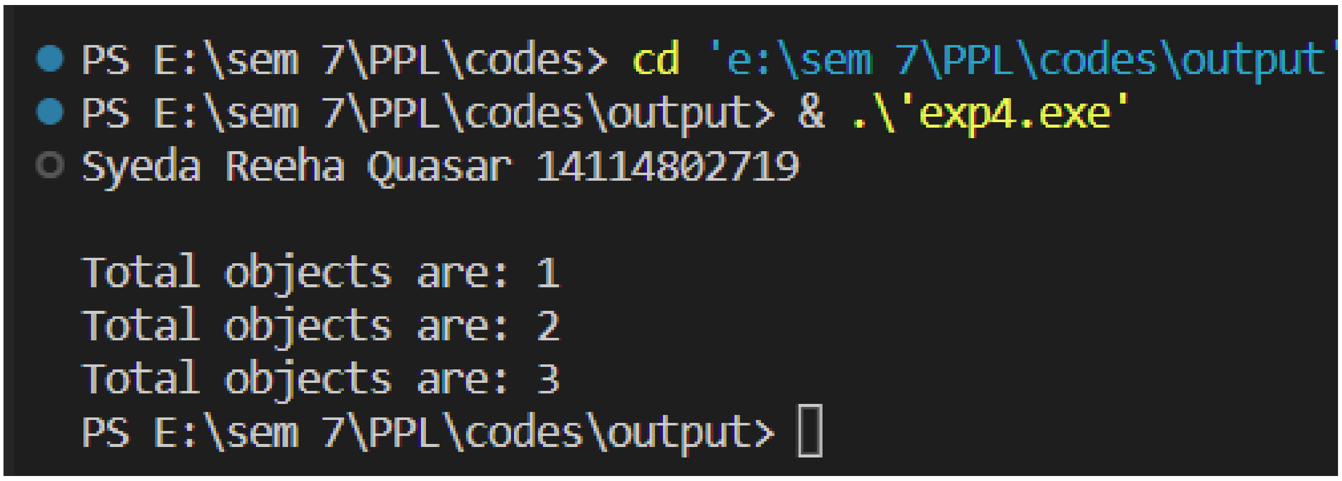
Experiment 4

Aim : WAP in C++ to count the no.s of object of a class with the help of static data member,funtion and constructor

Code :

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


class Counter
{


private:
    // static data member as count
    static int count;



public:

    // default constructor
    Counter()
    {
        count++;
    }

    // static member function
    static void Print()
    {
        cout << "\nTotal objects are: " << count;
    }
};



// count initialization with 0
int Counter ::count = 0;


int main()
{
    printf("Syeda Reeha Quasar 14114802719 \n");

    Counter OB1;
    OB1.Print();
    Counter OB2;
    OB2.Print();
    Counter OB3;
    OB3.Print();
    return 0;
}
```

Output :



```
PS E:\sem 7\PPL\codes> cd 'e:\sem 7\PPL\codes\output'
PS E:\sem 7\PPL\codes\output> & .\'exp4.exe'
Syeda Reeha Quasar 14114802719

Total objects are: 1
Total objects are: 2
Total objects are: 3
PS E:\sem 7\PPL\codes\output>
```

Viva Questions:

Q1. List various type of languages.
The different types of programming languages are:

- Procedural Programming Language
- Functional Programming Language
- Object-oriented Programming Language
- Scripting Programming Language
- Logic Programming Language

Q2. What are the issues for languages?
Issues are:

- Development on hardware technologies
- Simplicity and performance trade off
- Usage Area
- Potability
- Variety of Project's Design Patterns
- Safety and security

Q3. What is translation?
Translation is a programming language processor that modifies a computer program from one language to another. It takes a program written in the source program and modifies it into a machine program. It can find and detect the error.

Q4. What are different types of translation and their roles?
Generally, there are three types of translators -

- Compilers - A compiler takes the source code as a whole and translates it into object code all in one go.
- Interpreters - An interpreter translates source code into object code one instruction ata time.
- Assemblers - The purpose of an assembler is to translate assembly language intoobject code.

Experiment 5

Aim : WAP in C++ & Java to declare a class Time with data members mm for minutes, ss for seconds and hh for hours. Define a parameterize constructor to assign time to its objects. Add two time objects using member function and assign to third objects. Implement all possible cases of time

Code :

C++

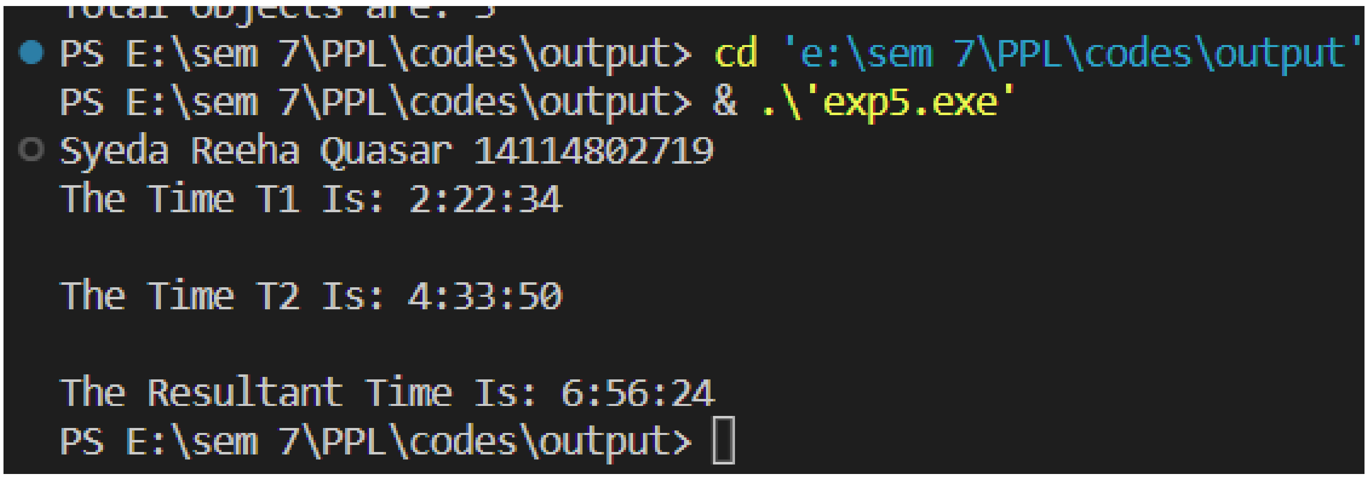
```
#include <iostream>
using namespace std;

class Time
{
    int hh, mm, ss;

public:
    Time() {}
    Time(int hh, int mm, int ss)
    {
        this->hh = hh;
        this->mm = mm;
        this->ss = ss;
    }
    void disp()
    {
        cout << hh << ":" << mm << ":" << ss;
    }
    void sum(Time t1, Time t2)
    {
        ss = t1.ss + t2.ss;
        mm = ss / 60;
        ss = ss % 60;
        mm = mm + t1.mm + t2.mm;
        hh = mm / 60;
        mm = mm % 60;
        hh = hh + t1.hh + t2.hh;
    }
};

int main()
{
    printf("Syeda Reeha Quasar 14114802719 \n");
    Time t1(2, 22, 34);
    cout << "The Time T1 Is: ";
    t1.disp();
    Time t2(4, 33, 50);
    cout << "\n\nThe Time T2 Is: ";
    t2.disp();
    Time t3;
    t3.sum(t1, t2);
    cout << "\n\nThe Resultant Time Is: ";
    t3.disp();
}
```

JAVA

```
import java.io.*;

class time {
    int hour, min, sec;

    public time(int h, int m, int s) {
        hour = h;
        min = m;
        sec = s;
    }

    public time() {
        hour = 0;
        min = 0;
        sec = 0;
    }

    public time sum(time t2) {
        time t3 = new time();
        t3.sec = sec + t2.sec;
        t3.min = min + t2.min;
        t3.hour = hour + t2.hour;
        if (t3.sec >= 60) {
            t3.min = t3.sec / 60;
            t3.sec = t3.sec % 60;
        }
        if (t3.min >= 60) {
            t3.hour = t3.min / 60;
            t3.min = t3.min % 60;
        }
        return (t3);
    }

    public void display() {
        System.out.println(hour + ":" + min + ":" + sec);
    }
}

public class Main {
    public static void main(String args[]) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Syeda Reeha Quasar 14114802719 \n\n");
        time t1 = new time(1, 56, 55);
        time t2 = new time(2, 00, 10);
        time t3 = new time();
        t3 = t1.sum(t2);
        System.out.print("Time 1:- ");
        t1.display();
        System.out.print("Time 2:- ");
        t2.display();
        System.out.print("Resultant time after addition is:- ");
        t3.display();
    }
}
```

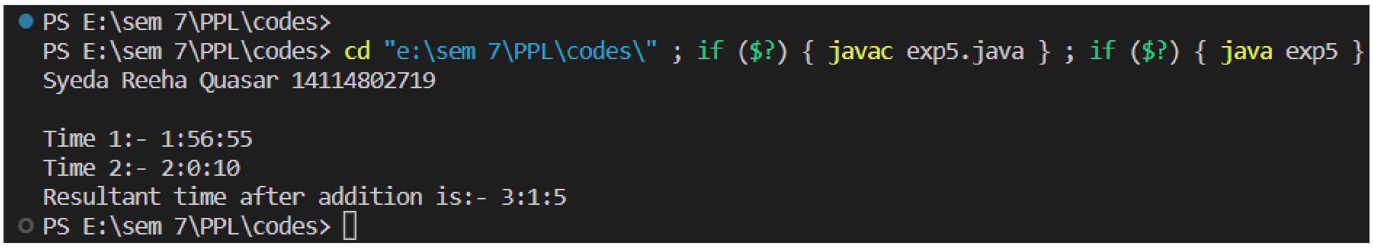
Output :

C++

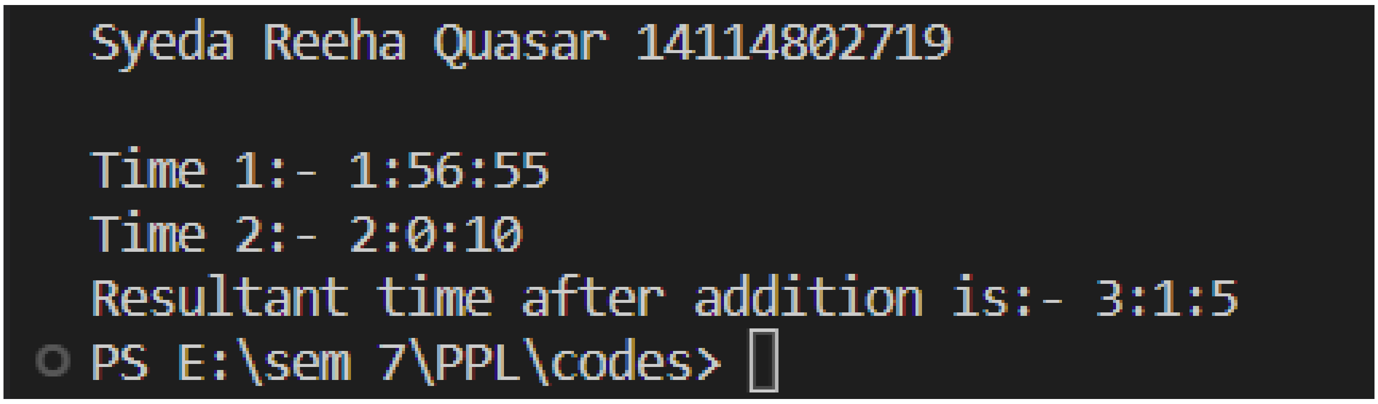


```
PS E:\sem 7\PPL\codes\output> cd 'e:\sem 7\PPL\codes\output'
PS E:\sem 7\PPL\codes\output> & .\'exp5.exe'
Syeda Reeha Quasar 14114802719
The Time T1 Is: 2:22:34

The Time T2 Is: 4:33:50

The Resultant Time Is: 6:56:24
PS E:\sem 7\PPL\codes\output>
```

JAVA



```
PS E:\sem 7\PPL\codes>
PS E:\sem 7\PPL\codes> cd "e:\sem 7\PPL\codes\" ; if ($?) { javac exp5.java } ; if ($?) { java exp5 }
Syeda Reeha Quasar 14114802719

Time 1:- 1:56:55
Time 2:- 2:0:10
Resultant time after addition is:- 3:1:5
PS E:\sem 7\PPL\codes>
```



```
Syeda Reeha Quasar 14114802719

Time 1:- 1:56:55
Time 2:- 2:0:10
Resultant time after addition is:- 3:1:5
PS E:\sem 7\PPL\codes>
```

Viva Questions:

Q1. Write any four important uses of programming languages.
A programming language is an artificial language that can be used to control the behaviour ofa machine, particularly a computer. Programming languages, like human languages, are defined through the use of syntactic and semantic rules, to determine structure and meaning respectively. Programming languages are used to facilitate communication about the task of organizing and manipulating information, and to express algorithms precisely.

Q2. Write the differences between lexical syntax and concrete syntax of the language. Lexical syntax for defining the rules for basic symbols involving identifiers, literals, punctuators and operators. Concrete syntax specifies the real representation of the programswith the help of lexical symbols like its alphabet.

Q3. List the design principle of imperative languages.
Imperative programming is a software development paradigm where functions are implicitly coded in every step required to solve a problem. In imperative programming, every operationis coded and the code itself specifies how the problem is to be solved, which means that
pre-coded models are not called on.

Q4. Write the differences between array and enumerated data types in imperativelanguages?
The main difference is that an array is a value and an enum is a type. And One main difference we can say that an array is a collection of other values (that is it contains other values, you can iterate through them or access individual ones by index), whereas an enumvalue is simply one atomic value.

Q5. Distinguish between dangling pointers and memory leakage.
Generally, daggling pointers arise when the referencing object is deleted or deallocated,
without changing the value of the pointers. In opposite to the dangling pointer, a memory leakoccurs when you forget to deallocate the allocated memory.

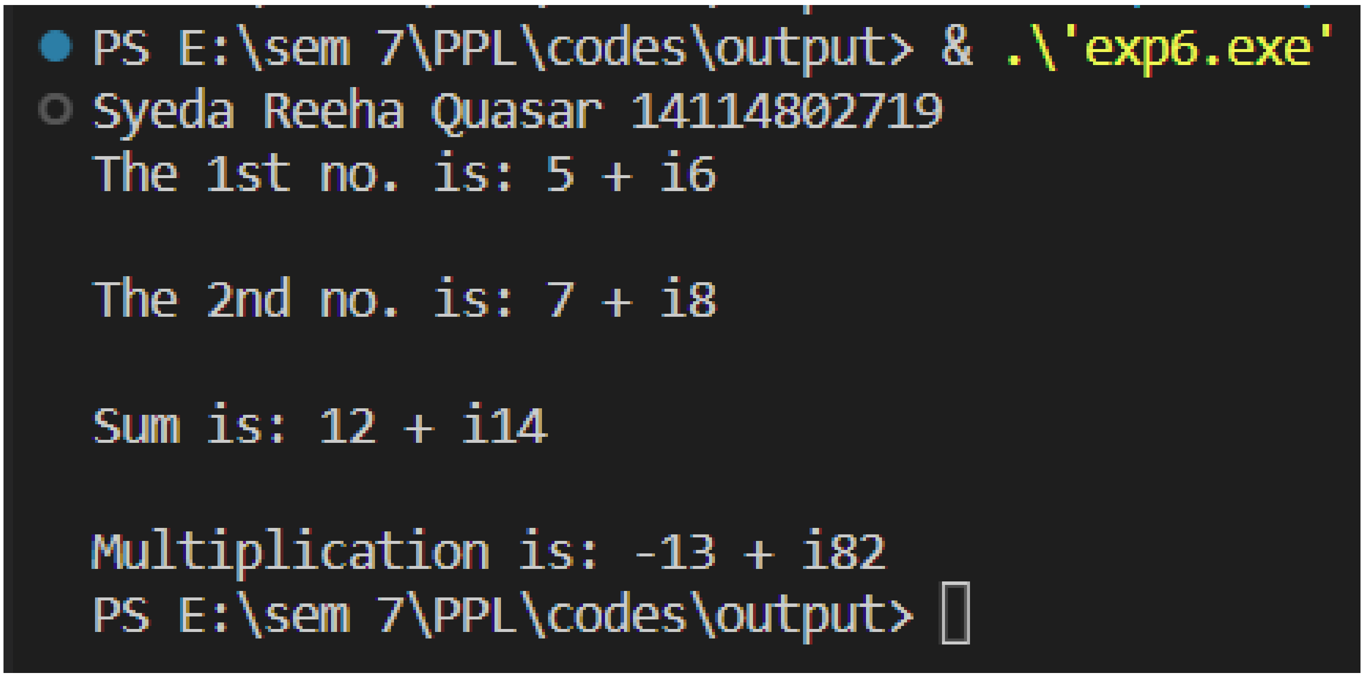
Experiment 6

Aim : WAP in C++ to define a class Complex to represents set of all complex numbers. Overload '+' operator to add two complex numbers using member function of the class andoverload '*' operator to multiply two complex numbers using friend function of the class complex.

Code :

```
#include <iostream>
using namespace std;
class Complex
{
   int real, img;

public:
   Complex(){};
   Complex(int i, int j)
   {
      real = i;
      img = j;
   }
   void show()
   {
      cout << real << " + i" << img;
   }
   Complex operator+(Complex obj)
   {
      Complex temp;
      temp.real = real + obj.real;
      temp.img = img + obj.img;
      return (temp);
   }
   Complex operator*(Complex);
};
Complex Complex::operator*(Complex c)
{
   double real1, real2;
   real1 = real;
   real2 = c.real;
   real = (real * c.real) - (img * c.img);
   img = (real1 * c.img) + (img * real2);
   Complex temp;
   temp.real = real;
   temp.img = img;
   return temp;
}
int main()
{
   printf("Syeda Reeha Quasar 14114802719 \n");

   Complex c1(5, 6), c2(7, 8), c3, c4;
   cout << "The 1st no. is: ";
   c1.show();
   cout << "\n\nThe 2nd no. is: ";
   c2.show();
   c3 = c1 + c2;
   cout << "\n\nSum is: ";
   c3.show();
   c4 = c1 * c2;
   cout << "\n\nMultiplication is: ";
   c4.show();
}
```

Output :



```
PS E:\sem 7\PPL\codes\output> & .\'exp6.exe'
Syeda Reeha Quasar 14114802719
The 1st no. is: 5 + i6

The 2nd no. is: 7 + i8

Sum is: 12 + i14

Multiplication is: -13 + i82
PS E:\sem 7\PPL\codes\output>
```

Viva Questions:

Q1. List the benefits of the modular development approach.
The advantages of the modular design include:

- Consistency. The fact that the engine allows all applications to run using the same base RAD applications brings consistency to the Service Manager application suite.
- Reduced Development Times.
- Flexibility.

Q2. Give some reasons why computer scientists and professional software developersshould study general concepts of language design and evaluation.
Reasons for Studying Concepts of Programming Languages

- Increased capacity to express ideas
- Improved ability to choose an appropriate language
- Increased ability to learn new languages
- A better understanding of implementation issues (i.e., how language constructs are implemented)

Q3. What constitutes a programming environment?
In a general sense, a programming environment combines hardware and software that allowsa developer to build applications. Developers typically work in integrated development environments or IDEs. These connect users with all the features necessary to write and testtheir code correctly.

Q4. Give an example of how aliasing deters reliability.
Aliasing makes it particularly difficult to understand, analyze and optimize programs.

Q5. How do type declaration statements affect the readability of programminglanguage?
Readability is a key element in any programming language and thus, very important. The addition of type declarations helps to improve code readability. This is because it makes iteasy to identify and differentiate different variables by data types. A program with
well-defined type declarations is easy to understand.

Experiment 7

Aim : Implement simple multi-threaded server to perform all mathematics operation parallelin Java.

Code :

SERVER

```
import java.io.*;
import java.net.*;

class Server {
    public static void main(String[] args) {
        ServerSocket server = null;
        try {
            server = new ServerSocket(1234);
            server.setReuseAddress(true);
            while (true) {
                Socket client = server.accept();
                System.out.println("New client connected" + client.getInetAddress().getHostAddress());
                ClientHandler clientSock = new ClientHandler(client);
                new Thread(clientSock).start();
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (server != null) {
                try {
                    server.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    private static class ClientHandler implements Runnable {
        private final Socket clientSocket;

        public ClientHandler(Socket socket) {
            this.clientSocket = socket;
        }

        public void run() {
            PrintWriter out = null;
            BufferedReader in = null;
            try {
                out = new PrintWriter(
                        clientSocket.getOutputStream(), true);
                in = new BufferedReader(
                        new InputStreamReader(
                                clientSocket.getInputStream()));
                String line;
                while ((line = in.readLine()) != null) {
                    String[] arr = line.split(" ");
                    int res = 0, p, q;
                    p = Integer.parseInt(arr[1]);
                    q = Integer.parseInt(arr[2]);
                    if (line.charAt(0) == '1') {
                        res = p + q;
                    }
                    if (line.charAt(0) == '2') {
                        res = p - q;
                    }
                    if (line.charAt(0) == '3') {
                        res = p * q;
                    }
                    if (line.charAt(0) == '4') {
                        res = p / q;
                    }
                    String text = "choice: " + String.valueOf(line.charAt(0)) + "\nAnd the Numbers are " +
p + " " + q;
                    System.out.printf(" Sent from the client: %s\n", text);
                    line = Integer.toString(res);
                    out.println(line);
                }
            } catch (IOException e) {
                e.printStackTrace();
            } finally {
                try {
                    if (out != null) {
                        out.close();
                    }
                    if (in != null) {
                        in.close();
                        clientSocket.close();
                    }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

CLIENT

```
import java.io.*;
import java.net.*;
import java.util.*;

class Client {
    public static void main(String[] args) {
        try (Socket socket = new Socket("localhost", 1234)) {
            PrintWriter out = new PrintWriter(
                    socket.getOutputStream(), true);
            BufferedReader in = new BufferedReader(new InputStreamReader(
                socket.getInputStream()));
            Scanner sc = new Scanner(System.in);
            String line = null;
            while (!"exit".equalsIgnoreCase(line)) {
                System.out.println("\nEnter Numbers: ");
                String a1, a2;
                a1 = sc.nextLine();
                a2 = sc.nextLine();
                System.out.println("1. Addition\n2. Subtraction\n3. Multiplication\n4. Division");
                String ch;
                System.out.println("\nEnter Choice: ");
                ch = sc.nextLine();
                line = ch + ' ' + a1 + ' ' + a2;
                out.println(line);
                out.flush();
                System.out.println("Server replied Result is: " + in.readLine());
            }
            sc.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output :

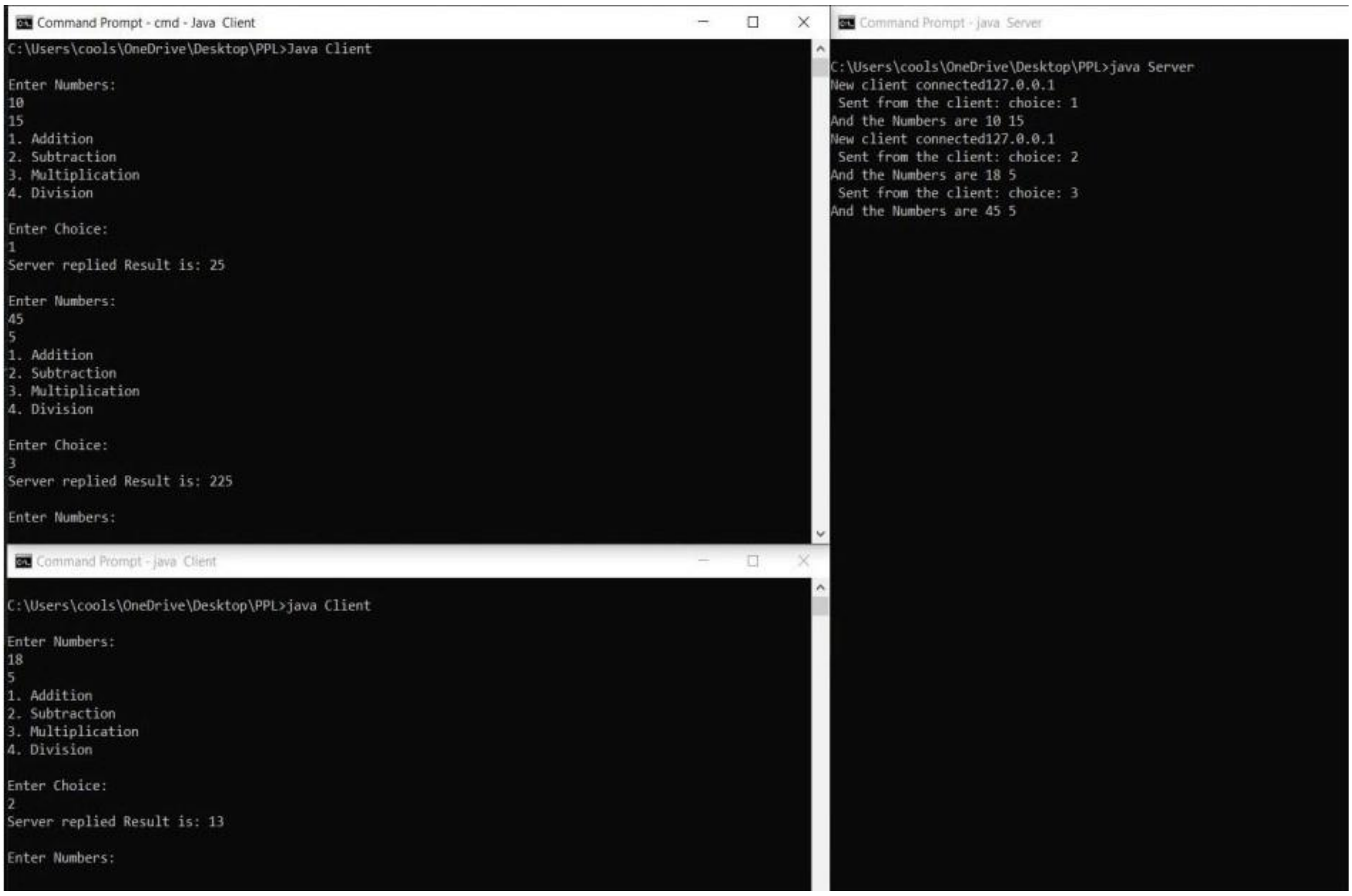
SERVER SIDE

```
*************
ARITHMETIC SERVER
*********************
Server is ready to accept inputs…
```

CLIENT SIDE

```
************
ARITHMETIC CLIENT
********************
Enter the host name to connectp4-221
Enter the inputs8
+ 3

Sever Says : 11

9
– 11

Sever Says : -2

7
* 9

Sever Says : 63

12
/ 3
```

Sever Says : 4



Command Prompt - cmd - Java Client

```
C:\Users\cools\OneDrive\Desktop\PPL>Java Client

Enter Numbers:
10
15
1. Addition
2. Subtraction
3. Multiplication
4. Division

Enter Choice:
1
Server replied Result is: 25

Enter Numbers:
45
5
1. Addition
2. Subtraction
3. Multiplication
4. Division

Enter Choice:
3
Server replied Result is: 225

Enter Numbers:
```

Command Prompt - java Client

```
C:\Users\cools\OneDrive\Desktop\PPL>java Client

Enter Numbers:
18
5
1. Addition
2. Subtraction
3. Multiplication
4. Division

Enter Choice:
2
Server replied Result is: 13

Enter Numbers:
```

Command Prompt - java Server

```
C:\Users\cools\OneDrive\Desktop\PPL>java Server
New client connected127.0.0.1
 Sent from the client: choice: 1
And the Numbers are 10 15
New client connected127.0.0.1
 Sent from the client: choice: 2
And the Numbers are 18 5
 Sent from the client: choice: 3
And the Numbers are 45 5
```

Viva Questions:

Q1. Write the uses of constructors and destructors in OOP.
Constructor is used to initializing an object of the class and assign values to datamembers corresponding to the class. While destructor is used to deallocate the memory of an object of a class. There can be multiple constructors for the same class.

Q2. Explain language evaluation criteria and the characteristics that affectthem.
Language Evaluation Criteria

- Criteria. Readability. Writability. Reliability. Cost.
- Affected by. Simplicity. Orthogonality. Data types and structures. Syntax.

Support for Abstraction. Type checking. Exception handling. Aliasing. Controlstructures. Expressivity.

Q3. What Is Backus-naur Form (bnf)?
Backus–Naur form or Backus normal form is a metasyntax notation for context-freegrammars, often used to describe the syntax of languages used in computing, suchas computer programming languages, document formats, instruction sets, and communication protocols.

Q4. What is the use of pointers?
Pointers are used to store and manage the addresses of dynamically allocated blocks of memory. Such blocks are used to store data objects or arrays of objects.

Q5. What is a Void pointer?
The void pointer is a pointer that is not associated with any data types. It points tosome data location in the storage. This means that it points to the address of variables. It is also called the general-purpose pointer.

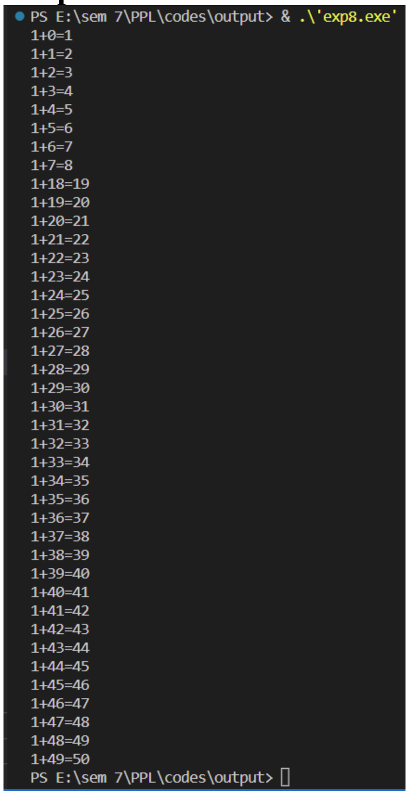
Experiment 8

Aim : Write a program in to prepare a list of 50 questions and their answers.

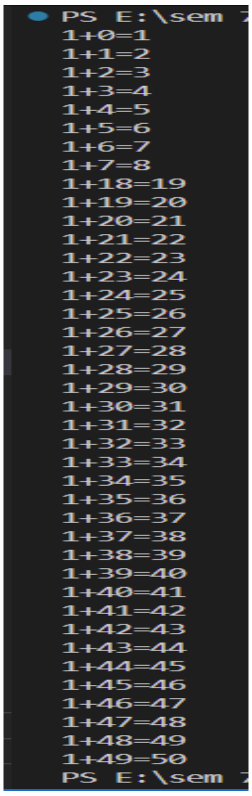
Code :

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

int main()
{
   string ques[50], ans[50];
   for (int i = 0, j = 1; i < 50; i++, j++)
   {
      stringstream s1, s2;
      s1 << i;
      s2 << j;
      ques[i] = "1+";
      ques[i] += s1.str();
      ans[i] = s2.str();
   }
   for (int i = 0; i < 50; i++)
      cout << ques[i] << "=" << ans[i] << endl;
   return 0;
}
```

Output :



```
PS E:\sem 7\PPL\codes\output> & .\'exp8.exe'
1+0=1
1+1=2
1+2=3
1+3=4
1+4=5
1+5=6
1+6=7
1+7=8
1+18=19
1+19=20
1+20=21
1+21=22
1+22=23
1+23=24
1+24=25
1+25=26
1+26=27
1+27=28
1+28=29
1+29=30
1+30=31
1+31=32
1+32=33
1+33=34
1+34=35
1+35=36
1+36=37
1+37=38
1+38=39
1+39=40
1+40=41
1+41=42
1+42=43
1+43=44
1+44=45
1+45=46
1+46=47
1+47=48
1+48=49
1+49=50
PS E:\sem 7\PPL\codes\output> 
```

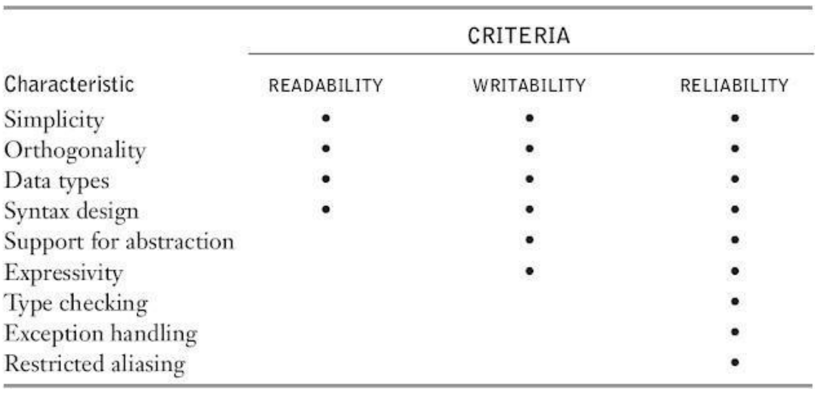


```
PS E:\sem
1+0=1
1+1=2
1+2=3
1+3=4
1+4=5
1+5=6
1+6=7
1+7=8
1+18=19
1+19=20
1+20=21
1+21=22
1+22=23
1+23=24
1+24=25
1+25=26
1+26=27
1+27=28
1+28=29
1+29=30
1+30=31
1+31=32
1+32=33
1+33=34
1+34=35
1+35=36
1+36=37
1+37=38
1+38=39
1+39=40
1+40=41
1+41=42
1+42=43
1+43=44
1+44=45
1+45=46
1+46=47
1+47=48
1+48=49
1+49=50
PS E:\sem
```

Viva Questions:

Q1. Explain language evaluation criteria and the characteristics that affect them.
The most prominent language evaluation criteria are:

- Readability
- Writability
- Reliability



	CRITERIA		
Characteristic	READABILITY	WRITABILITY	RELIABILITY
Simplicity	•	•	•
Orthogonality	•	•	•
Data types	•	•	•
Syntax design	•	•	•
Support for abstraction		•	•
Expressivity		•	•
Type checking			•
Exception handling			•
Restricted aliasing			•

Q2. What is the use of functions? How are actual parameters different from formal parameters?
In computer science, Backus–Naur form or Backus normal form (BNF) is a notation technique for context-free grammars, often used to describe the syntax of languages used in computing, such as computer programming languages, document formats, instruction sets and communication protocols. They are applied wherever exact descriptions of languages are needed: for instance, in official language specifications, in manuals, and in textbooks on programming language theory.

Q3. What is a void pointer?
A void pointer is a pointer that has no associated data type with it. A void pointer can hold address of any type and can be typcasted to any type.

Experiment 9

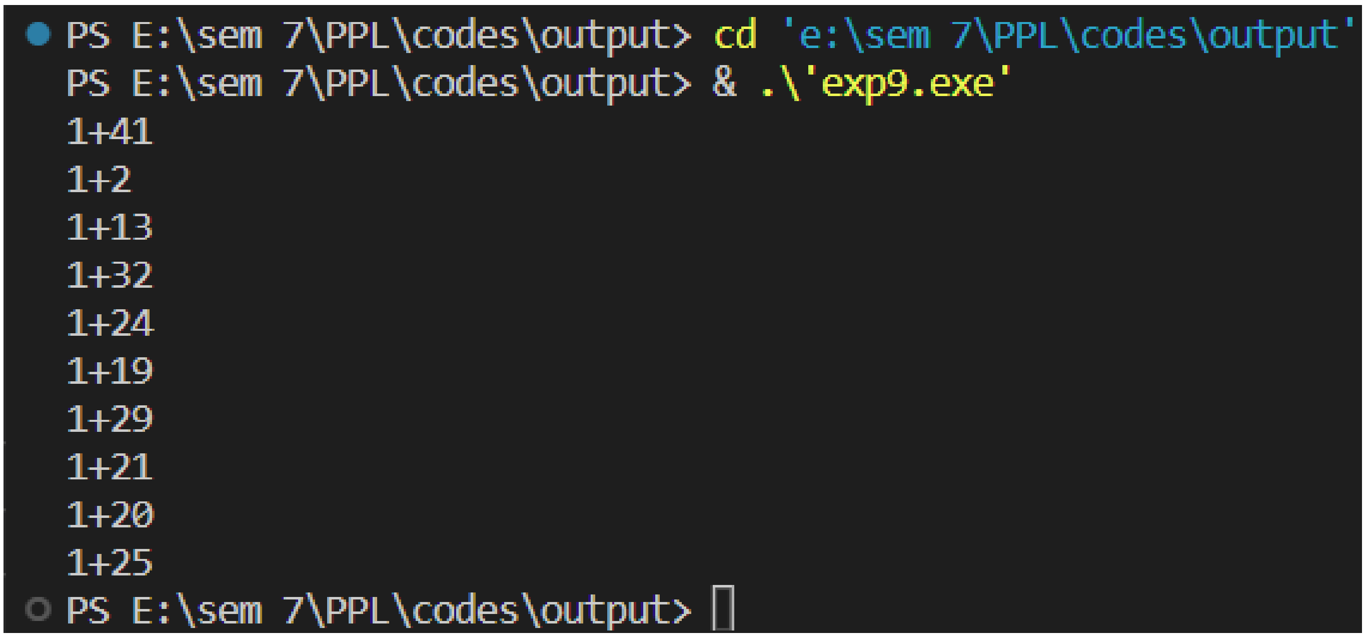
Aim : Write a program to display 10 questions at random out of exp.8-50 questions (do not display the answer of these questions to the user now).

Code :

```
#include <iostream>
#include <string>
#include <sstream>
#include <ctime>
#include <stdlib.h>
using namespace std;

void generateSetOfNumbers(int arr[]);

int main()
{
    string ques[50], ans[50];
    int r[10];
    for (int i = 0, j = 1; i < 50; i++, j++)
    {
        stringstream s1, s2;
        s1 << i;
        s2 << j;
        ques[i] = "1+";
        ques[i] += s1.str();
        ans[i] = s2.str();
    }

    srand(time(0));
    generateSetOfNumbers(r);
    for (int i = 0; i < 10; i++)
    {
        int x = r[i];
        cout << ques[x] << endl;
    }
    return 0;
}

void generateSetOfNumbers(int arr[])
{
    int p[50] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49};
    for (int i = 49; i > 0; --i)
    {
        int j = rand() % i;
        int temp = p[i];
        p[i] = p[j];
        p[j] = temp;
    }
    for (int i = 0; i < 10; ++i)
        arr[i] = p[i];
```

}

Output :



```
PS E:\sem 7\PPL\codes\output> cd 'e:\sem 7\PPL\codes\output'
PS E:\sem 7\PPL\codes\output> & .\'exp9.exe'
1+41
1+2
1+13
1+32
1+24
1+19
1+29
1+21
1+20
1+25
PS E:\sem 7\PPL\codes\output>
```

Viva Questions:

Q1. List various types of pointers.

- NULL pointer
- Dangling pointer
- Generic pointer
- Wild pointer
- Complex pointer

Q2. What is the use of functions? How are actual parameters different from formal parameters?

- Functions provide a high degree of modularity for your application and also provide better code reusability and ease of maintenance.
- Actual parameters are used in function calling statement. Formal parameters are used in function definition statement.

Q3. How are threads created?

- In Call by value method original value is not modified whereas, in Call by reference method, the original value is modified.
- In Call by value, a copy of the variable is passed whereas in Call by reference, a variable itself is passed.
- In Call by value, actual and formal arguments will be created in different memory locations whereas in Call by reference, actual and formal arguments will be created in the same memory location.

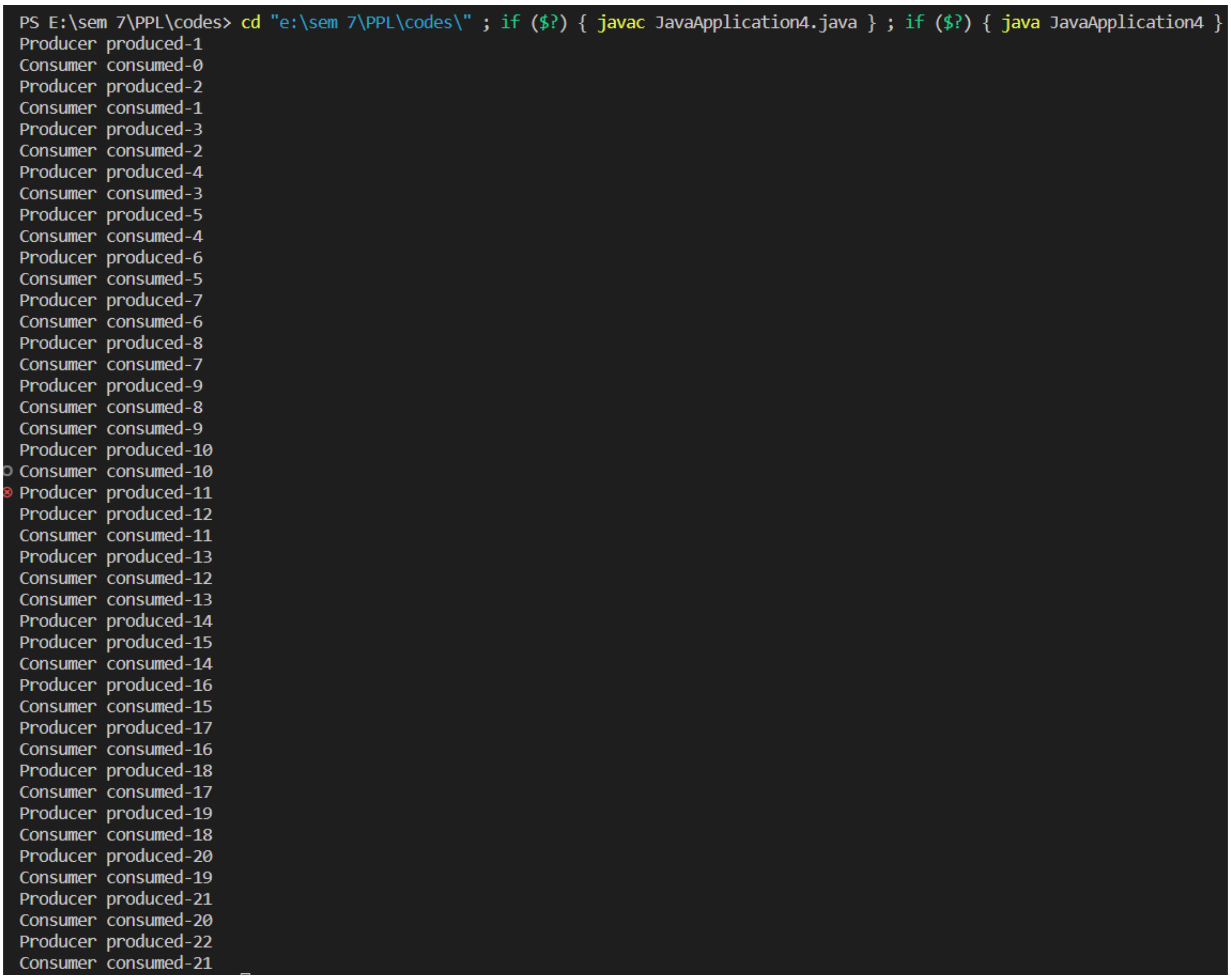
Experiment 10

Aim : Implement producer-consumer problem using threads.

Code :

```
import java.util.LinkedList;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;

public class JavaApplication4 {
    public static void main(String[] args) throws InterruptedException {
        final BlockingQueue<Integer> queue = new LinkedBlockingQueue<>();
        final PC pc = new PC();
        Thread t1 = new Thread(new Runnable() {
            public void run() {
                try {
                    pc.produce(queue);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });
        Thread t2 = new Thread(new Runnable() {
            public void run() {
                try {
                    pc.consume(queue);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });
        t1.start();
        t2.start();
        t1.join();
        t2.join();
    }

    public static class PC {
        public void produce(BlockingQueue<Integer> queue) throws
InterruptedException {
            int value = 0;
            while (true) {
                queue.add(value++);
                System.out.println("Producer produced-" + value);
                Thread.sleep(1000);
            }
        }

        public void consume(BlockingQueue<Integer> queue) throws
InterruptedException {
            while (true) {
                int val = queue.take();
                System.out.println("Consumer consumed-" + val);
                Thread.sleep(1000);
            }
        }
    }
}
```

Output :



```
PS E:\sem 7\PPL\codes> cd "e:\sem 7\PPL\codes\" ; if ($?) { javac JavaApplication4.java } ; if ($?) { java JavaApplication4 }
Producer produced-1
Consumer consumed-0
Producer produced-2
Consumer consumed-1
Producer produced-3
Consumer consumed-2
Producer produced-4
Consumer consumed-3
Producer produced-5
Consumer consumed-4
Producer produced-6
Consumer consumed-5
Producer produced-7
Consumer consumed-6
Producer produced-8
Consumer consumed-7
Producer produced-9
Consumer consumed-8
Consumer consumed-9
Producer produced-10
Consumer consumed-10
Producer produced-11
Producer produced-12
Consumer consumed-11
Producer produced-13
Consumer consumed-12
Consumer consumed-13
Producer produced-14
Producer produced-15
Consumer consumed-14
Producer produced-16
Consumer consumed-15
Producer produced-17
Consumer consumed-16
Producer produced-18
Consumer consumed-17
Producer produced-19
Consumer consumed-18
Producer produced-20
Consumer consumed-19
Producer produced-21
Consumer consumed-20
Producer produced-22
Consumer consumed-21
```

Viva Questions:

Q1. What is the role of producer and consumer in producer-consumer problem?
We have a buffer of fixed size. A producer can produce an item and place it in the buffer. A consumer can pick items and consume them. We need to ensure that when a producer is placing an item in the buffer, then at the same time consumer should not consume any item. In this problem, buffer is the critical section.

Q2. What is a semaphore?

- Semaphore is simply a variable that is non-negative and shared between threads. A semaphore is a signaling mechanism, and a thread that is waiting on a semaphore can be signaled by another thread.
  It uses two atomic operations,
  1) wait, and 2) signal for the process synchronization.

- A semaphore either allows or disallows access to the resource, which depends on how it is set up.

Q3. How are threads created?

- Java lets you create threads in one of two ways:
- By implementing the Runnable Interface
- By extending the Thread class

BEYOND THE SYLLABUS PRACTICALS

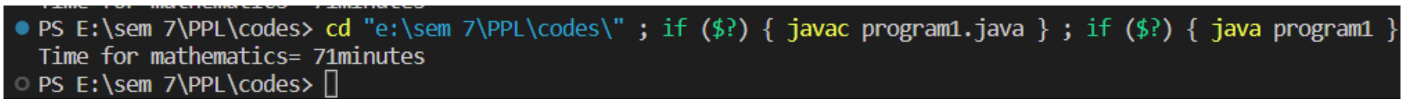
Experiment 1

Aim : There are 200 questions on a 3 hr examination. Among these questions are 50 mathematics problems. It is suggested that twice as much time be spent on each maths problem as for each other question. WAP which calculates how many minutes should be spent on mathematics problems.

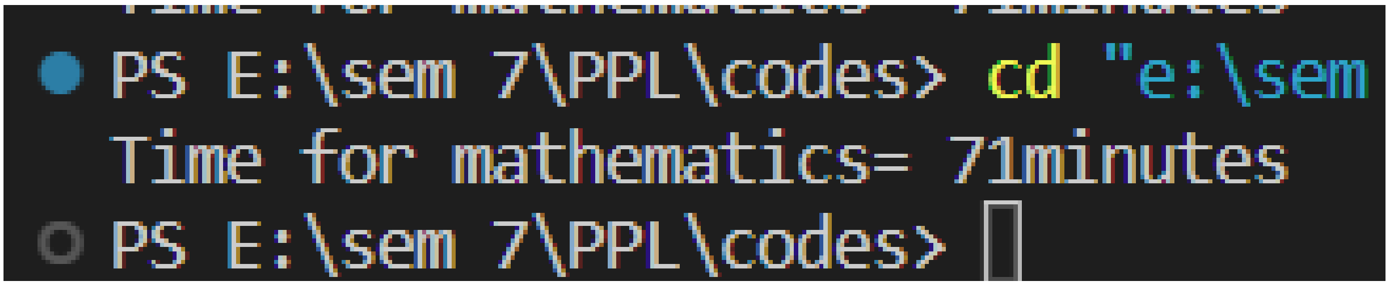
Code :

```
public class program1 {
    public static void main(String[] args) {
        final int time = 3 * 60 * 60;
        int mathsQues = 50, ques = 150, t;
        t = time / (ques + 2 * mathsQues);
        System.out.println("Time for mathematics= " + (mathsQues * 2 * t) / 60 +
"minutes");
    }
}
```

Output :



```
● PS E:\sem 7\PPL\codes> cd "e:\sem 7\PPL\codes\" ; if ($?) { javac program1.java } ; if ($?) { java program1 }
  Time for mathematics= 71minutes
○ PS E:\sem 7\PPL\codes> ▯
```



```
● PS E:\sem 7\PPL\codes> cd "e:\sem
  Time for mathematics= 71minutes
○ PS E:\sem 7\PPL\codes> ▯
```

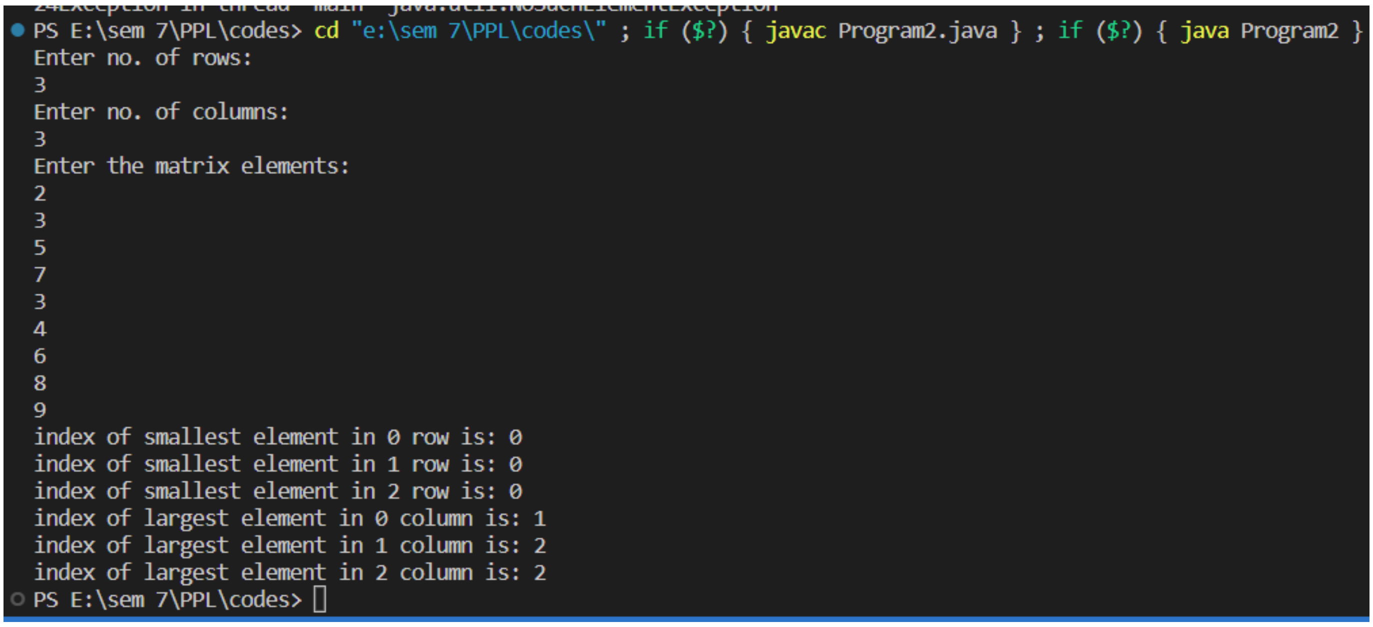
Experiment 2

Aim : User enters the elements in a m x n matrix, where m is the number of rows and n is the number of columns. Values of m and n are also entered by the user. Now WAP in C and JAVA which find out the position of the element which is smalfilest in the row and largest in the column.

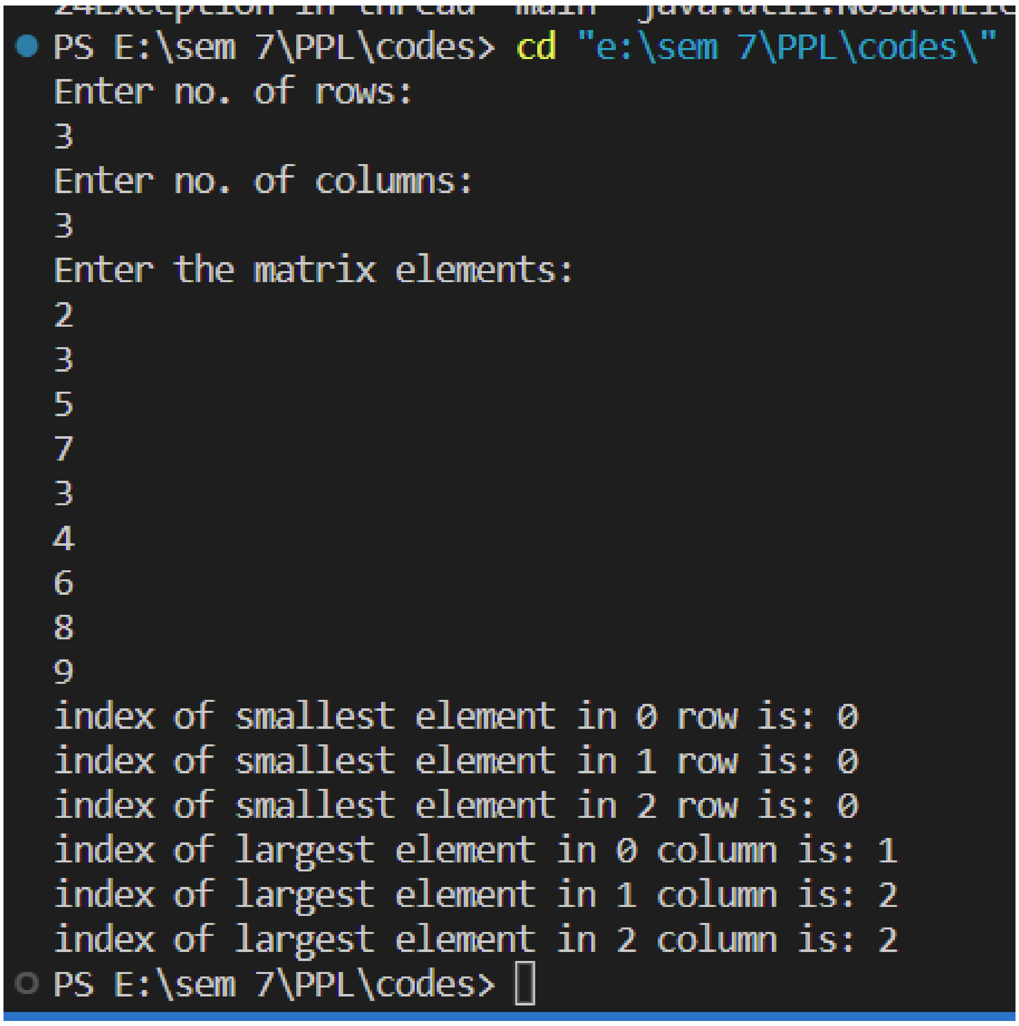
Code :

```
import java.util.Scanner;

public class Program2 {
    public static void main(String[] args) {
        int m, n;
        System.out.println("Enter no. of rows:");
        Scanner sc = new Scanner(System.in);
        m = sc.nextInt();
        System.out.println("Enter no. of columns:");
        n = sc.nextInt();
        int A[][] = new int[m][n];
        System.out.println("Enter the matrix elements:");
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                A[i][j] = sc.nextInt();
            }
        }
        int X = 0, y = 0, min = 100, max = 0;
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                if (min > A[i][j]) {
                    min = A[i][j];
                    y = j;
                }
            }
            System.out.println("index of smallest element in " + i + " row is: " + y);
        }
        for (int j = 0; j < m; j++) {
            for (int i = 0; i < n; i++) {
                if (max < A[i][j]) {
                    max = A[i][j];
                    X = i;
                }
            }
            System.out.println("index of largest element in " + j + " column is: " + X);
        }
    }
}
```

Output :



```
24Exception in thread "main" java.util.NoSuchElementException
PS E:\sem 7\PPL\codes> cd "e:\sem 7\PPL\codes\" ; if ($?) { javac Program2.java } ; if ($?) { java Program2 }
Enter no. of rows:
3
Enter no. of columns:
3
Enter the matrix elements:
2
3
5
7
3
4
6
8
9
index of smallest element in 0 row is: 0
index of smallest element in 1 row is: 0
index of smallest element in 2 row is: 0
index of largest element in 0 column is: 1
index of largest element in 1 column is: 2
index of largest element in 2 column is: 2
PS E:\sem 7\PPL\codes> 
```



```
24Exception in thread "main" java.util.NoSuchEle
PS E:\sem 7\PPL\codes> cd "e:\sem 7\PPL\codes\"
Enter no. of rows:
3
Enter no. of columns:
3
Enter the matrix elements:
2
3
5
7
3
4
6
8
9
index of smallest element in 0 row is: 0
index of smallest element in 1 row is: 0
index of smallest element in 2 row is: 0
index of largest element in 0 column is: 1
index of largest element in 1 column is: 2
index of largest element in 2 column is: 2
PS E:\sem 7\PPL\codes> 
```

Experiment 3

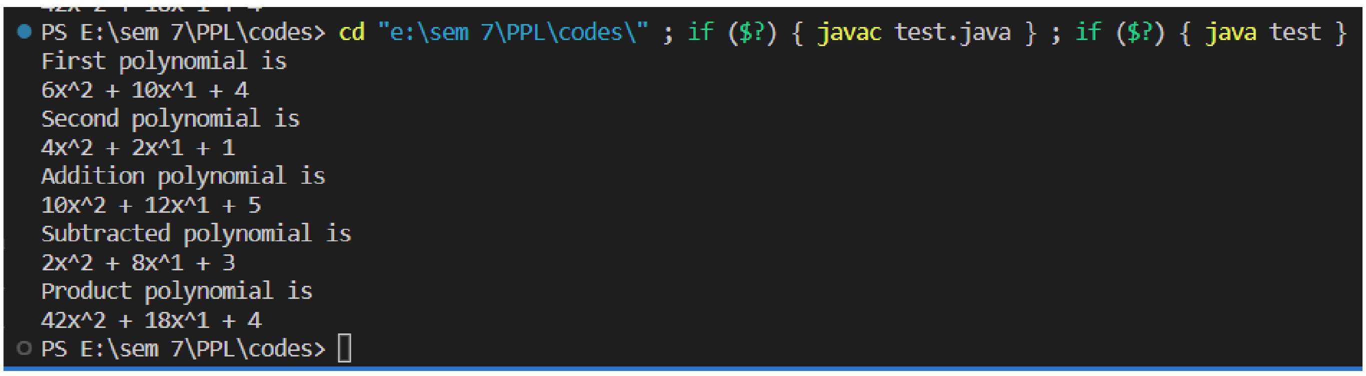
Aim : Two polynomials are entered by the user in the form of: ax2 + bx + c where the powers of x can be any integer value and a, b & c are constants. Now WAP in C and JAVA which calculates the sum, product and difference of the two polynomials.

Code :

JAVA

```
import java.util.Scanner;

class test {
    public static int[] multiply(int A[], int B[], int m, int n) {
        int[] prod = new int[m + n - 1];
        for (int i = 0; i < m + n - 1; i++) {
            prod[i] = 0;
        }
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                prod[i + j] += A[i] * B[j];
            }
        }
        return prod;
    }

    public static void multiply(int A[], int B[], int n) {
        int[] prod = new int[2 * n - 1];
        for (int i = 0; i < prod.length; i++) {
            prod[i] = 0;
        }
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                prod[i + j] += A[i] * B[j];
            }
        }
        printPoly(prod, n);
    }

    public static void addition(int A[], int B[], int n) {
        int[] add = new int[n];
        for (int i = 0; i < add.length; i++) {
            add[i] = A[i] + B[i];
        }
        printPoly(add, n);
    }
    public static void subtract(int A[], int B[], int n) {
        int[] sub = new int[n];
        for (int i = 0; i < sub.length; i++) {
            sub[i] = A[i] - B[i];
        }
        printPoly(sub, n);
    }

    static void printPoly(int poly[], int n) {
        for (int i = n - 1; i >= 0; i--) {
            System.out.print(poly[i]);
            if (i != 0) {
                System.out.print("x^" + i + " + ");
                continue;
            }
        }
        System.out.println();
    }

    public static void main(String[] args) {
        int A[] = { 4, 10, 6 };
        int B[] = { 1, 2, 4 };
        int n = A.length;
        System.out.println("First polynomial is ");
        printPoly(A, n);
        System.out.println("Second polynomial is ");
        printPoly(B, n);
        System.out.println("Addition polynomial is \t");
        addition(A, B, n);
        System.out.println("Subtracted polynomial is \t");
        subtract(A, B, n);
        System.out.println("Product polynomial is \t");
        multiply(A, B, n);
    }
}
```

Output :



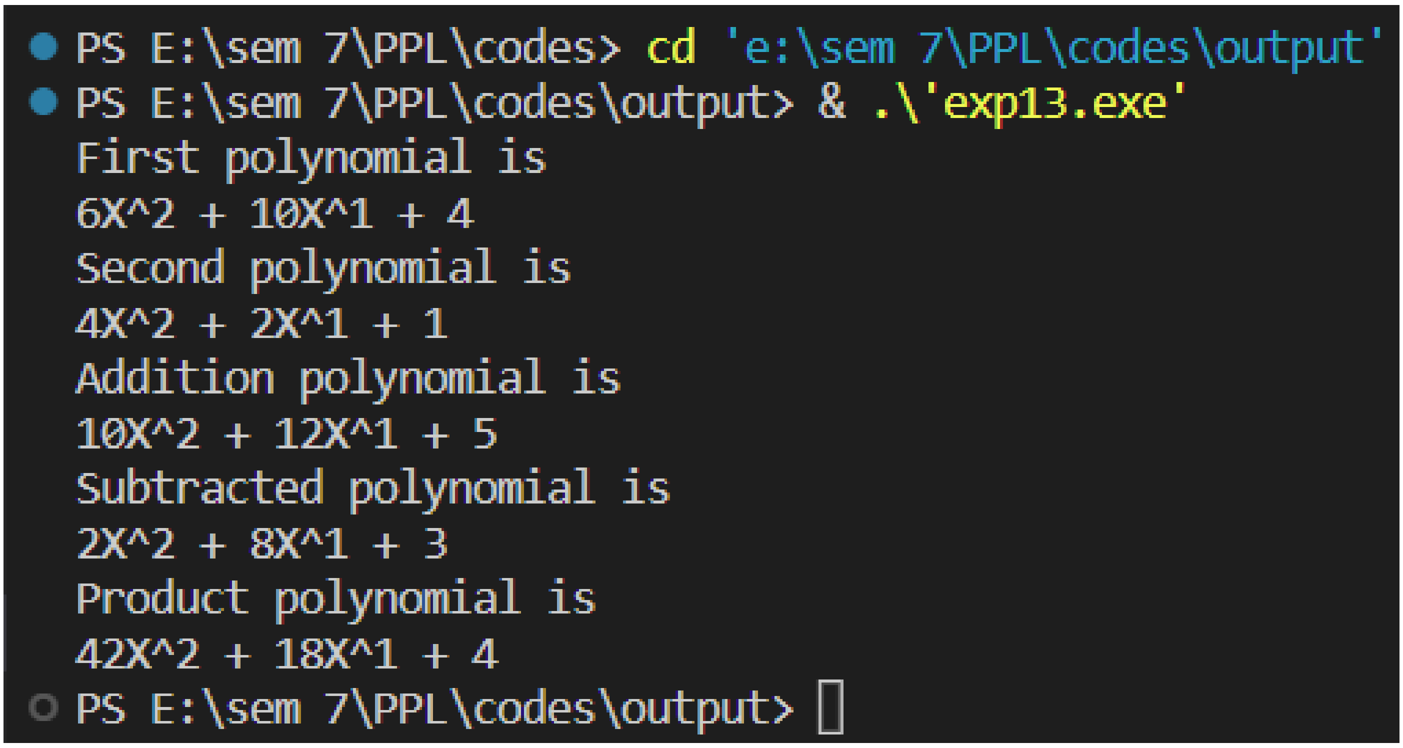
```
PS E:\sem 7\PPL\codes> cd "e:\sem 7\PPL\codes\" ; if ($?) { javac test.java } ; if ($?) { java test }
First polynomial is
6x^2 + 10x^1 + 4
Second polynomial is
4x^2 + 2x^1 + 1
Addition polynomial is
10x^2 + 12x^1 + 5
Subtracted polynomial is
2x^2 + 8x^1 + 3
Product polynomial is
42x^2 + 18x^1 + 4
PS E:\sem 7\PPL\codes>
```

C++

```
#include <math.h>
#include <stdio.h>

#define MAX 17

void init(int p[]);
void read(int p[]);
void print(int p[]);
void addition(int p1[], int p2[]);
void multiply(int p1[], int p2[]);
void subtract(int p1[], int p2[]);

int main()
{
   int A[3] = {4, 10, 6}, B[3] = {1, 2, 4};
   printf("First polynomial is \n");
   print(A);
   printf("Second polynomial is \n");
   print(B);
   printf("Addition polynomial is \n");
   addition(A, B);
   printf("Subtracted polynomial is \n");
   subtract(A, B);
   printf("Product polynomial is \n");
   multiply(A, B);
   return 0;
}

void print(int p[])
{
   int i;
   for (i = 2; i >= 0; i--)
   {
      printf("%d", p[i]);
      if (i != 0)
         printf("X^%d + ", i);
   }
   printf("\n");
}

void addition(int p1[], int p2[])
{
   int i;
   int p3[3];
   for (i = 0; i < 3; i++)
```

```
        p3[i] = p1[i] + p2[i];
    print(p3);
}

void subtract(int p1[], int p2[])
{
    int i;
    int p3[3];
    for (i = 0; i < 3; i++)
        p3[i] = p1[i] - p2[i];
    print(p3);
}

void multiply(int p1[], int p2[])
{
    int i, j;
    int p3[5];
    for (i = 0; i < 3; i++)
        p3[i] = 0;
    for (i = 0; i < 3; i++)
        for (j = 0; j < 3; j++)
            p3[i + j] = p3[i + j] + p1[i] * p2[j];
    print(p3);
}
```

Output :



```
PS E:\sem 7\PPL\codes> cd 'e:\sem 7\PPL\codes\output'
PS E:\sem 7\PPL\codes\output> & .\'exp13.exe'
First polynomial is
6X^2 + 10X^1 + 4
Second polynomial is
4X^2 + 2X^1 + 1
Addition polynomial is
10X^2 + 12X^1 + 5
Subtracted polynomial is
2X^2 + 8X^1 + 3
Product polynomial is
42X^2 + 18X^1 + 4
PS E:\sem 7\PPL\codes\output>
```

Experiment 4

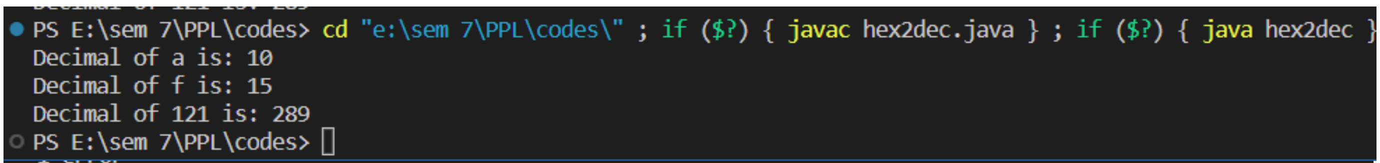
Aim : The hexadecimal digits are the ordinary, base-10 digits '0' through '9' plus the letters 'A' through 'F'. In the hexadecimal system, these digits represent the values 0 through 15, respectively. Write a function in JAVA and C named hexValue that uses a switch statement to find the hexadecimal value of a given character. The character is a parameter to the function, and its hexadecimal value is the return value of the function. You should count lower case letters 'a' through 'f' as having the same value as the corresponding upper case letters. If the parameter is not one of the legal hexadecimal digits, return -1 as the value of the function.

Code :

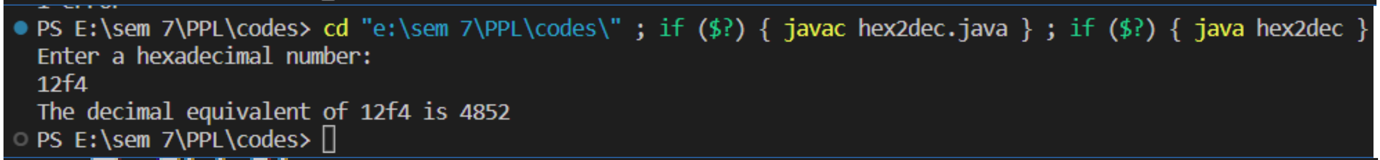
JAVA

```
public class hex2dec {
   public static void main(String[] args) {
   String hex;
   long dec;
   int i;
   TextIO.put("Enter a hexadecimal number: ");
   hex = TextIO.getlnWord();
   dec = 0;
   for ( i = 0; i < hex.length(); i++ ) {
   int digit = hexValue( hex.charAt(i) ); if (digit == -1) {
   TextIO.putln("Error: Input is not a hexadecimal number.");
   return; }
   dec = 16*dec + digit; }
   TextIO.putln("Base-10 value: " + dec);
   }

   static int hexValue(char ch) {
   switch (ch) {
   case '0':   return 0;
   case '1':   return 1;
   case '2':   return 2;
   case '3':   return 3;
   case '4':   return 4;
   case '5':   return 5;
   case '6':   return 6;
   case '7':   return 7;
   case '8':   return 8;
   case '9':   return 9;
   case 'a':
   case 'A':   return 10;
   case 'b':
   case 'B':   return 11;
   case 'c':
   case 'C':   return 12;
   case 'd':
   case 'D':   return 13;
   case 'e':
   case 'E':   return 14;
   case 'f':
   case 'F':   return 15;
   default:    return -1;
   }} }
```

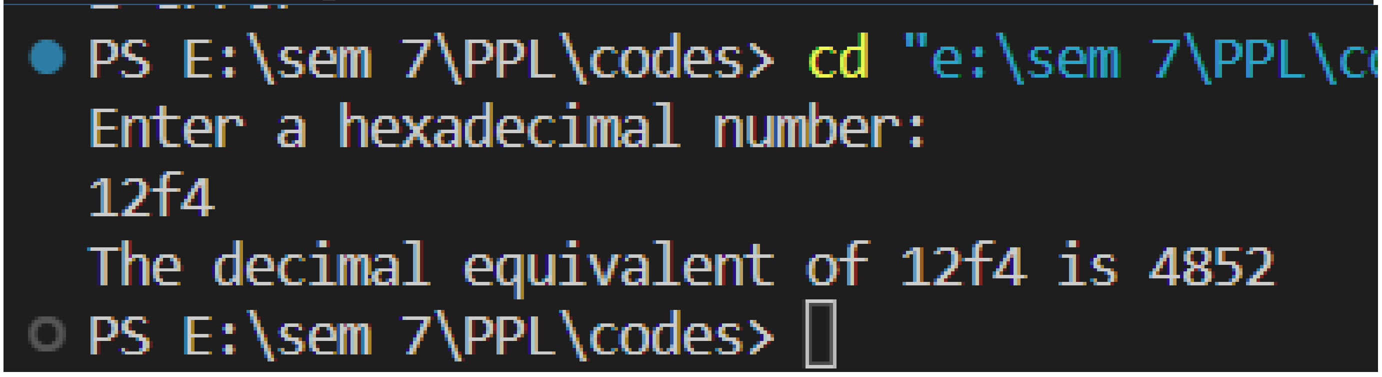
Output :



```
● PS E:\sem 7\PPL\codes> cd "e:\sem 7\PPL\codes\" ; if ($?) { javac hex2dec.java } ; if ($?) { java hex2dec }
  Decimal of a is: 10
  Decimal of f is: 15
  Decimal of 121 is: 289
○ PS E:\sem 7\PPL\codes> 
```



```
● PS E:\sem 7\PPL\codes> cd "e:\sem 7\PPL\codes\" ; if ($?) { javac hex2dec.java } ; if ($?) { java hex2dec }
  Enter a hexadecimal number:
  12f4
  The decimal equivalent of 12f4 is 4852
○ PS E:\sem 7\PPL\codes> 
```

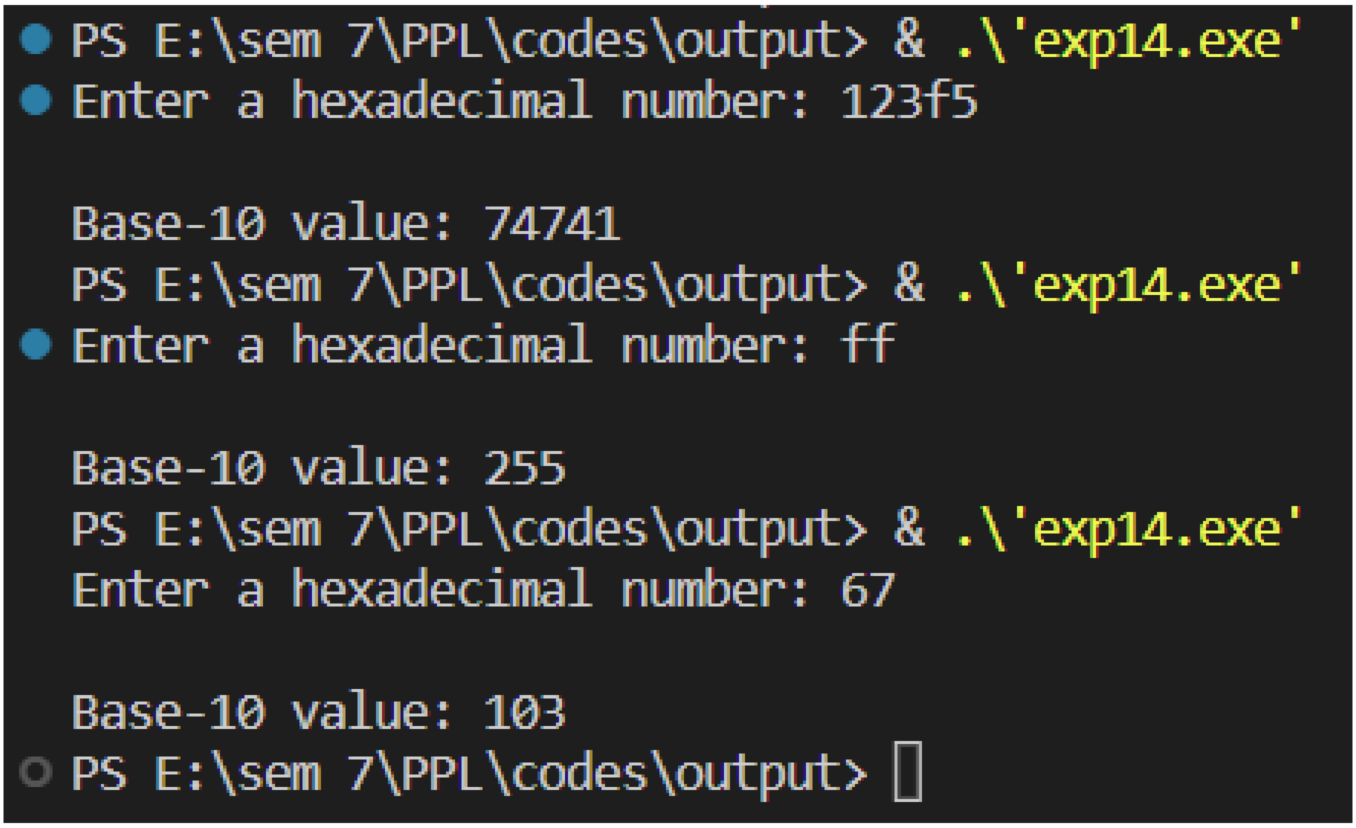


```
● PS E:\sem 7\PPL\codes> cd "e:\sem 7\PPL\c
  Enter a hexadecimal number:
  12f4
  The decimal equivalent of 12f4 is 4852
○ PS E:\sem 7\PPL\codes> 
```

C++

```
#include <iostream>
using namespace std;
int hexValue(char ch); int main(){
string hex; long dec; int i;
cout<<"Enter a hexadecimal number: "; cin>>hex;
cout<<endl; dec = 0;
for ( i = 0; i < hex.length(); i++ ) { int digit = hexValue( hex.at(i) );
if (digit == -1) { cout<<"Error: Input is not a hexadecimal number."; return 0; }
dec = 16*dec + digit; }
cout<<"Base-10 value: "<<dec; }

int hexValue(char ch) {
switch (ch) {
case '0':       return 0;
case '1':       return 1;
case '2':       return 2;
case '3':       return 3;
case '4':       return 4;
case '5':       return 5;
case '6':       return 6;
case '7':       return 7;
case '8':       return 8;
case '9':       return 9;
case 'a':
case 'A':       return 10;
case 'b':
case 'B':       return 11;
case 'c':
case 'C':       return 12;
case 'd':
case 'D':       return 13;
case 'e':
case 'E':       return 14;
case 'f':
case 'F':       return 15;
default:        return -1; }}
```

Output :



```
PS E:\sem 7\PPL\codes\output> & .\'exp14.exe'
Enter a hexadecimal number: 123f5

Base-10 value: 74741
PS E:\sem 7\PPL\codes\output> & .\'exp14.exe'
Enter a hexadecimal number: ff

Base-10 value: 255
PS E:\sem 7\PPL\codes\output> & .\'exp14.exe'
Enter a hexadecimal number: 67

Base-10 value: 103
PS E:\sem 7\PPL\codes\output>
```

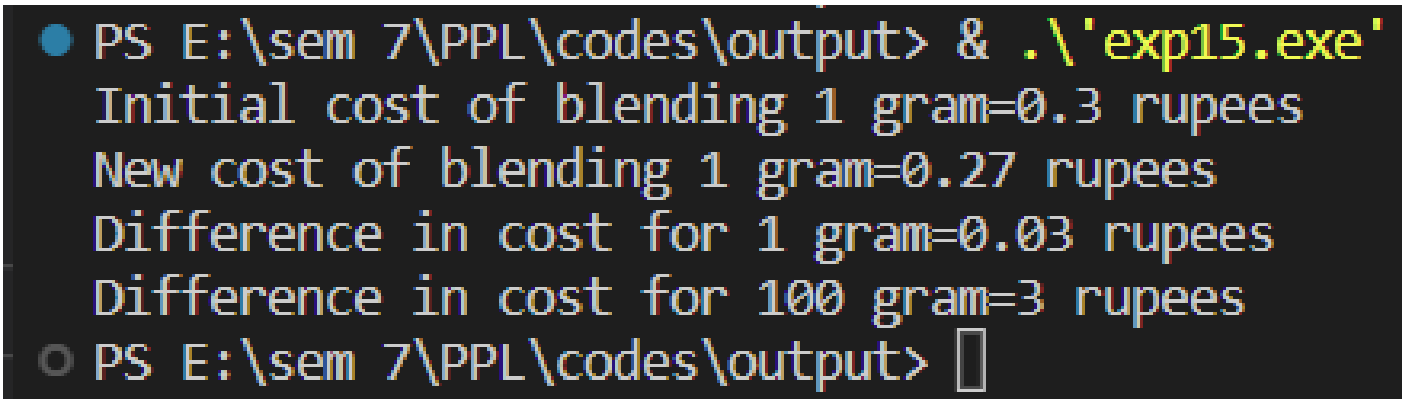
Experiment 5

Aim : A coffee shop blends 2 kinds of coffee, putting in 2 parts of a 33p a gm. grade to 1 part of a 24p a gm. If the mixture is changed to 1 part of the 33p a gm. to 2 parts of the filess expensive grade .WAP which calculates that how much will the shop save in blending 100 gms.

Code :

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

int costOfBlending(int x, int y);

int main()
{
    cout << "Initial cost of blending 1 gram=" << costOfBlending(2, 1) / 100.00 << "
rupees" << endl;
    cout << "New cost of blending 1 gram=" << costOfBlending(1, 2) / 100.00 << "
rupees" << endl;
    cout << "Difference in cost for 1 gram=" << (costOfBlending(2, 1) -
costOfBlending(1, 2)) / 100.00 << " rupees" << endl;
    cout << "Difference in cost for 100 gram=" << costOfBlending(2, 1) -
costOfBlending(1, 2) << " rupees" << endl;
    return 0;
}

int costOfBlending(int x, int y)
{
    int costOfNGram = x * 33 + y * 24;
    int costOfOneGram = (x * 33 + y * 24) / (x + y);
    return costOfOneGram;
}
```

Output :



```
PS E:\sem 7\PPL\codes\output> & .\'exp15.exe'
Initial cost of blending 1 gram=0.3 rupees
New cost of blending 1 gram=0.27 rupees
Difference in cost for 1 gram=0.03 rupees
Difference in cost for 100 gram=3 rupees
PS E:\sem 7\PPL\codes\output>
```

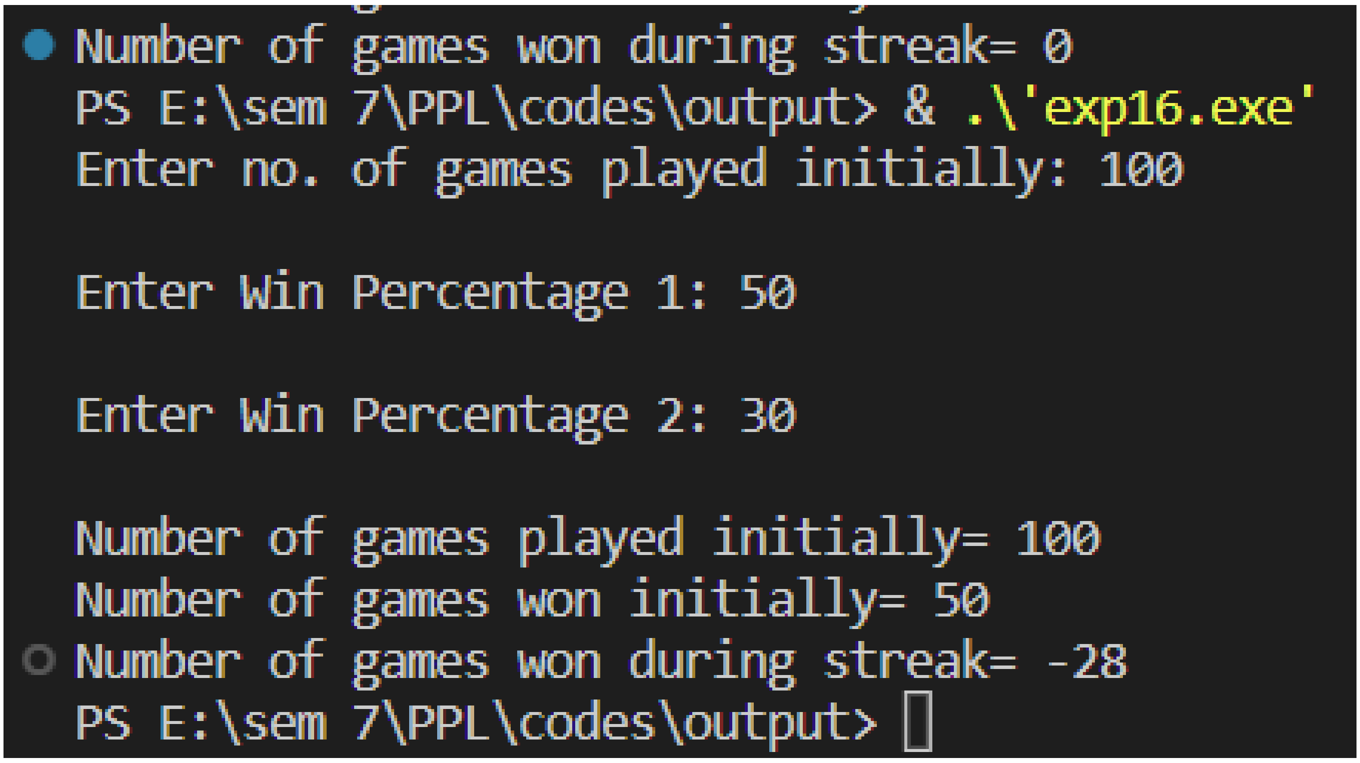
Experiment 6

Aim : In June a baseball team that played 60 games had won 30% of its game played. After a phenomenal winning streak this team raised its average to 50%. WAP which calculates how many games must the team have won in a row to attain this average.

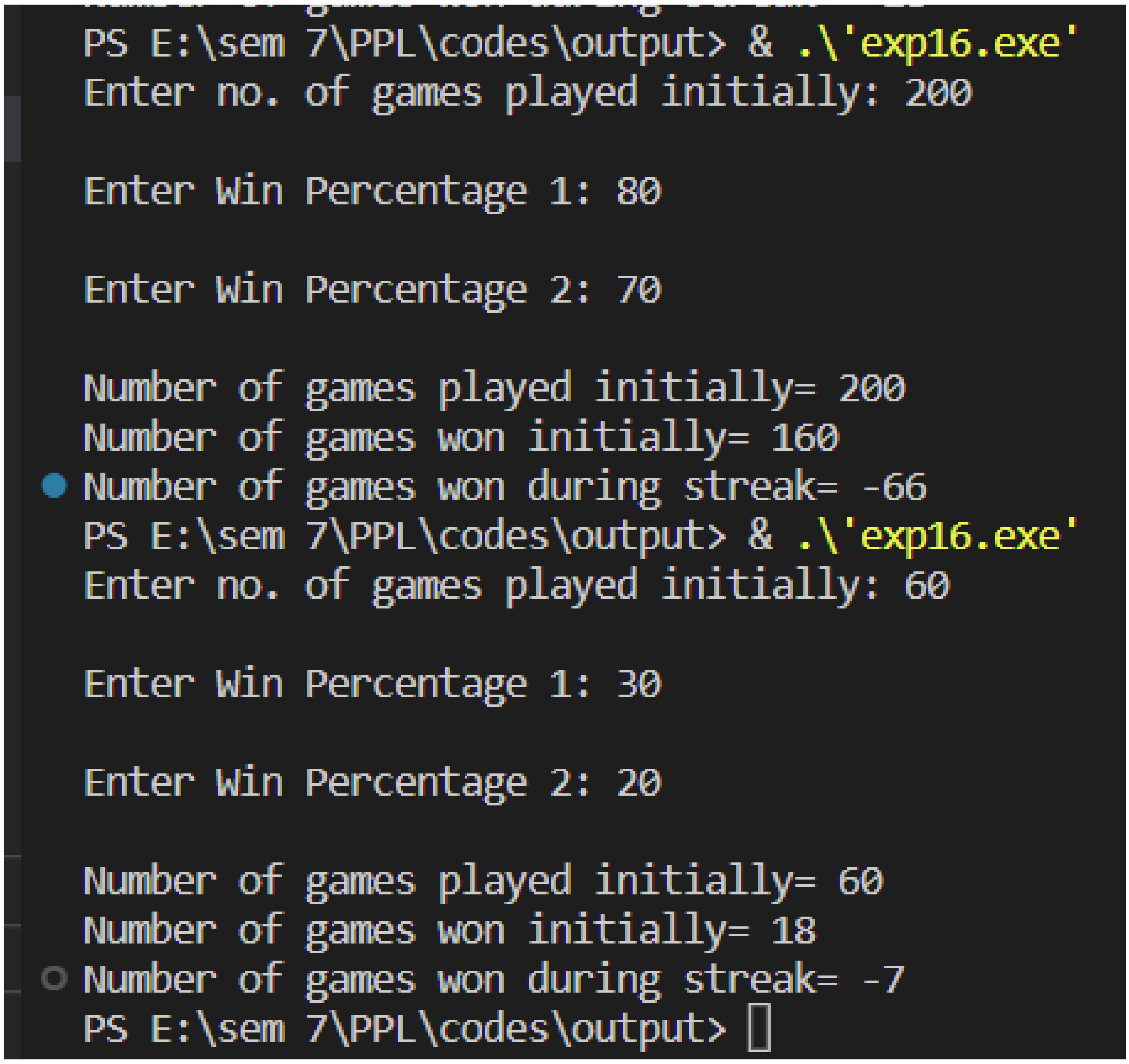
Code :

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

int winsNeeded(int gamesPlayed1, int winPercentage1, int winPercentage2);

int main()
{
   int a, b, c;

   cout << "Enter no. of games played initially: ";
   cin >> a;
   cout << endl;

   cout << "Enter Win Percentage 1: ";
   cin >> b;
   cout << endl;

   cout << "Enter Win Percentage 2: ";
   cin >> c;
   cout << endl;

   cout << "Number of games played initially= " << a << endl;
   cout << "Number of games won initially= " << (a * b) / 100 << endl;
   cout << "Number of games won during streak= " << winsNeeded(a, b, c) << endl;
   return 0;
}

int winsNeeded(int gamesPlayed1, int winPercentage1, int winPercentage2)
{
   int gamesWon1, gamesWon2;

   gamesWon1 = (gamesPlayed1 * winPercentage1) / 100;
   gamesWon2 = (gamesPlayed1 * winPercentage2 - 100 * gamesWon1) / (100 -
winPercentage2);

   return gamesWon2;
}
```

Output :



```
Number of games won during streak= 0
PS E:\sem 7\PPL\codes\output> & .\'exp16.exe'
Enter no. of games played initially: 100

Enter Win Percentage 1: 50

Enter Win Percentage 2: 30

Number of games played initially= 100
Number of games won initially= 50
Number of games won during streak= -28
PS E:\sem 7\PPL\codes\output> 
```



```
PS E:\sem 7\PPL\codes\output> & .\'exp16.exe'
Enter no. of games played initially: 200

Enter Win Percentage 1: 80

Enter Win Percentage 2: 70

Number of games played initially= 200
Number of games won initially= 160
Number of games won during streak= -66
PS E:\sem 7\PPL\codes\output> & .\'exp16.exe'
Enter no. of games played initially: 60

Enter Win Percentage 1: 30

Enter Win Percentage 2: 20

Number of games played initially= 60
Number of games won initially= 18
Number of games won during streak= -7
PS E:\sem 7\PPL\codes\output> 
```

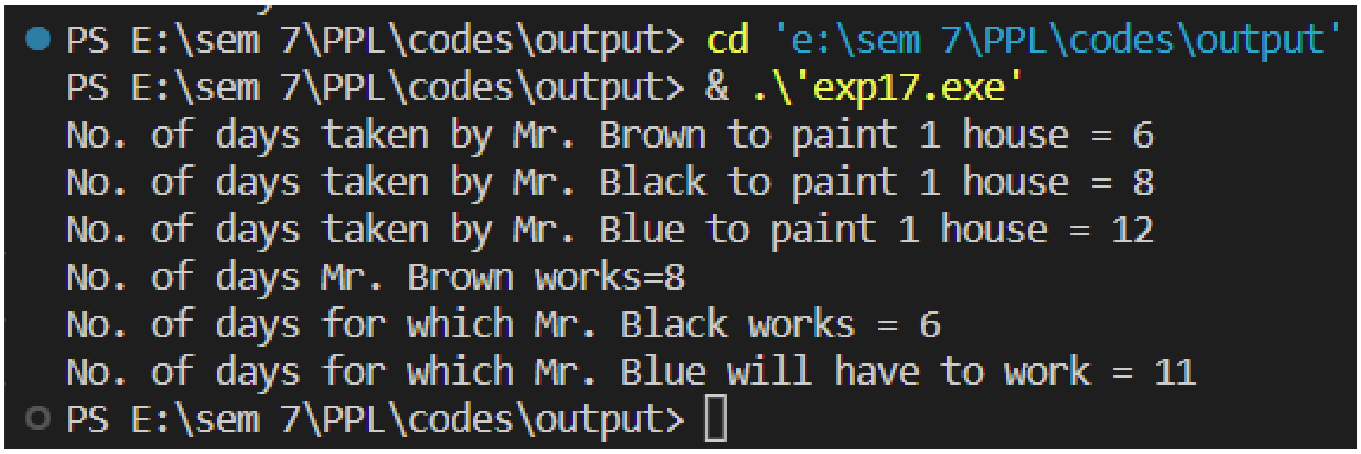
Experiment 7

Aim : A company contracts to paint 3 houses. Mr. Brown can paint a house in 6 days while Mr. Black would take 8 days and Mr. Blue 12 days. After 8 days Mr. Brown goes on vacation and Mr. Black begins to work for a period of 6 days. WAP which calculates how many days will it take Mr. Blue to complete the contract.

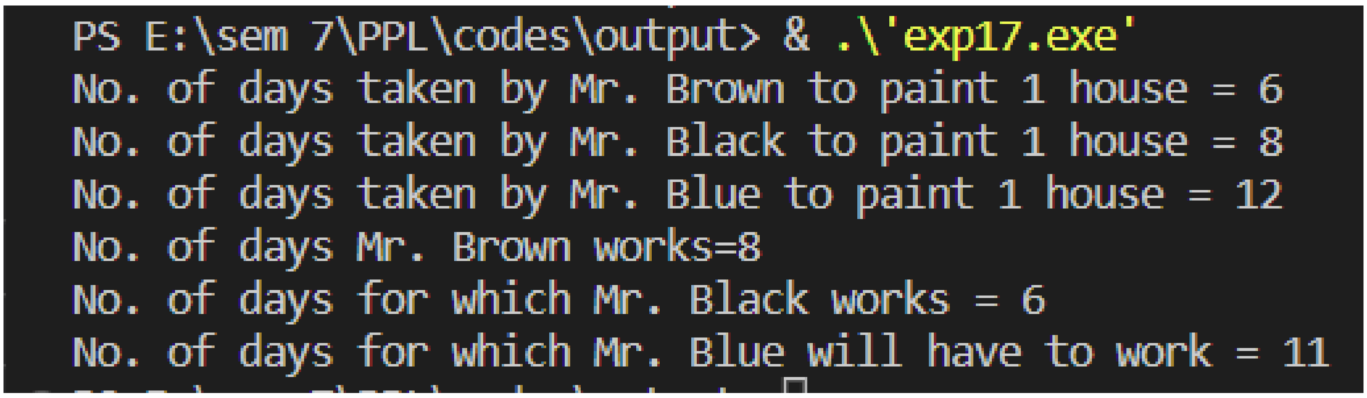
Code :

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

int gcd(int a, int b);

int main()
{
   float result;
   int mul;
   cout << "No. of days taken by Mr. Brown to paint 1 house=6" << endl;
   cout << "No. of days taken by Mr. Black to paint 1 house=8" << endl;
   cout << "No. of days taken by Mr. Blue to paint 1 house=12" << endl;
   cout << "No. of days Mr. Brown works=8" << endl;
   cout << "No. of days for which Mr. Black works=6" << endl;

   mul = gcd(gcd(6, 8), 12);
   result = ((3 * mul - ((6 * mul) / 8.0) - ((8 * mul) / 6.0)) * 12.0) / mul;

   cout << "No. of days for which Mr. Blue will have to work=" << result << endl;

   return 0;
}

int gcd(int a, int b)
{
   if (a == 0)
      return b;
   else if (b == 0)
      return a;
   else if (a == b)
      return a;
   else if (a > b)
      return gcd(a - b, b);

   return gcd(a, b - a);
}
```

Output :



```
PS E:\sem 7\PPL\codes\output> cd 'e:\sem 7\PPL\codes\output'
PS E:\sem 7\PPL\codes\output> & .\'exp17.exe'
No. of days taken by Mr. Brown to paint 1 house = 6
No. of days taken by Mr. Black to paint 1 house = 8
No. of days taken by Mr. Blue to paint 1 house = 12
No. of days Mr. Brown works=8
No. of days for which Mr. Black works = 6
No. of days for which Mr. Blue will have to work = 11
PS E:\sem 7\PPL\codes\output>
```



```
PS E:\sem 7\PPL\codes\output> & .\'exp17.exe'
No. of days taken by Mr. Brown to paint 1 house = 6
No. of days taken by Mr. Black to paint 1 house = 8
No. of days taken by Mr. Blue to paint 1 house = 12
No. of days Mr. Brown works=8
No. of days for which Mr. Black works = 6
No. of days for which Mr. Blue will have to work = 11
```

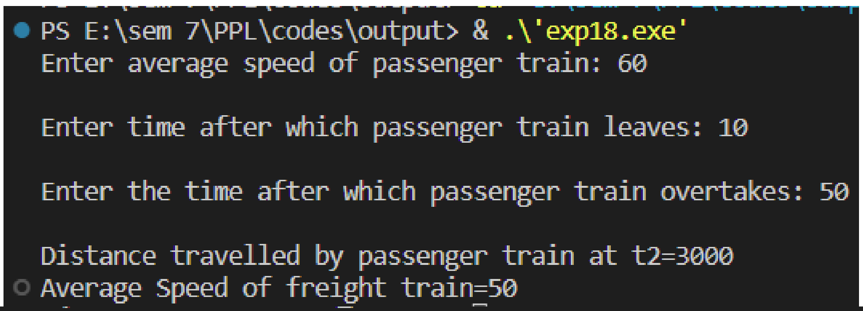
Experiment 8

Aim : 2 hours after a freight train leaves Delhi a passenger train leaves the same station traveling in the same direction at an average speed of 16 km/hr. After traveling 4 hrs the passenger train overtakes the freight train. WAP which calculates the average speed of the freight train.

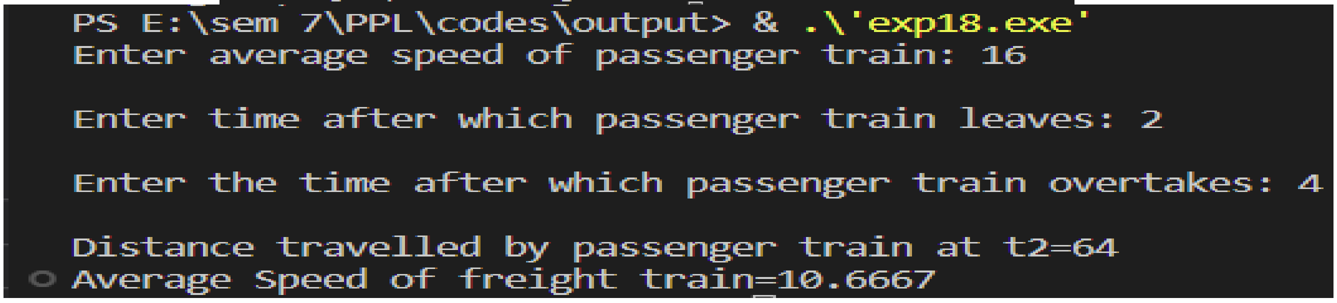
Code :

```
#include <iostream>
using namespace std;
int main()
{
   float avg_speed, t1, t2, answer;

   cout << "Enter average speed of passenger train: ";
   cin >> avg_speed;
   cout << endl;

   cout << "Enter time after which passenger train leaves: ";
   cin >> t1;
   cout << endl;

   cout << "Enter the time after which passenger train overtakes: ";
   cin >> t2;
   cout << endl;

   cout << "Distance travelled by passenger train at t2=" << avg_speed * t2 << endl;
   cout << "Average Speed of freight train=" << (avg_speed * t2) / (t1 + t2) << endl;
}
```

Output :



```
PS E:\sem 7\PPL\codes\output> & .\'exp18.exe'
Enter average speed of passenger train: 60

Enter time after which passenger train leaves: 10

Enter the time after which passenger train overtakes: 50

Distance travelled by passenger train at t2=3000
Average Speed of freight train=50
```



```
PS E:\sem 7\PPL\codes\output> & .\'exp18.exe'
Enter average speed of passenger train: 16

Enter time after which passenger train leaves: 2

Enter the time after which passenger train overtakes: 4

Distance travelled by passenger train at t2=64
Average Speed of freight train=10.6667
```