**PRINCIPAL OF PROGRAMMING LAB**

**(ETCS – 458)**

**Faculty name:** Ms. RUCHI **Student name:** Ayush Pandey

**Roll No.:** 45014802718

**Semester:** 8th Semester

**Group:** 8-C-9



Maharaja Agrasen Institute of Technology

PSP Area, Sector – 22, Rohini, New Delhi – 110085



**MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY**

**VISION**

To nurture young minds in a learning environment of high academic value and imbibe spiritual and ethical values with technological and management competence.

**MISSION**

**The Institute shall endeavor to incorporate the following basic missions in the teaching methodology:**

**Engineering Hardware – Software Symbiosis**

Practical exercises in all Engineering and Management disciplines shall be carried out by Hardware equipment as well as the related software enabling deeper understanding of basic concepts and encouraging inquisitive nature.

**Life – Long Learning**

The Institute strives to match technological advancements and encourage students to keep updating their knowledge for enhancing their skills and inculcating their habit of continuous learning.

**Liberalization and Globalization**

The Institute endeavors to enhance technical and management skills of students so that they are intellectually capable and competent professionals with Industrial Aptitude to face the challenges of globalization.

**Diversification**

The Engineering, Technology and Management disciplines have diverse fields of studies with different attributes. The aim is to create a synergy of the above attributes by encouraging analytical thinking.

**Entrepreneurship**

The Institute strives to develop potential Engineers and Managers by enhancing their skills and research capabilities so that they become successful entrepreneurs and responsible citizens.



**MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY**

**COMPUTER SCIENCE AND ENGINEERING DEPARTMENT**

**VISION**

To produce “Critical Thinkers of Innovative Technology”.

**MISSION**

To foster an open, multidisciplinary and highly collaborative research environment for producing world-class engineers capable of providing innovative solutions to real life problems and fulfil societal needs.

**PRACTICAL RECORD**

**PAPER CODE : ETCS-458**

**Name of the student : Ayush Pandey**

**University Roll No. : 45014802718**

**Branch : CSE**

**Group : 8C-9**

**PRACTICAL DETAILS**

1. Experiments according to PPL lab syllabus prescribed by GGSIPU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ExpNo.** | **Experiment Name** | **Date Of Performance** | **Total Marks** | **Signature with Date** |
| 1. | To implement all major functions of string.h in single C program using switch case to select specific function from user choice (like strlen, strcat, strcpy, strcmp, strrev). |  |  |  |
| 2. | Write a program in C to reverse a linked list iterative and recursive. |  |  |  |
| 3. | Write a program in C to implement iterative Towers of Hanoi. |  |  |  |
| 4. | WAP in C++ to count the numbers of object of a class with the help of static data member, function and constructor. |  |  |  |
| 5. | 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. |  |  |  |
| 6. | 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 and overload ‘\*’ operator to multiply two complex numbers using friend function of the class complex. |  |  |  |
| 7. | Implement simple multi-threaded server to perform all mathematical operations parallel in Java. |  |  |  |
| 8. | Write a program in to prepare a list of 50 questions and their answers. |  |  |  |
| 9. | Write a program to display 10 questions at random out of exp. 8-50 questions. |  |  |  |
| 10. | Implement producer-consumer problem using threads. |  |  |  |
| 11. | 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. |  |  |  |
| 12. | User enters the elements in an 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. |  |  |  |
| 13. | 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. |  |  |  |
| 14. | 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. |  |  |  |
| 15. | 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. |  |  |  |
| 16. | 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. |  |  |  |
| 17 | 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. |  |  |  |
| 18. | 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:** To implement all major functions of string.h in single C program using switch case to select specific function from user choice (like strlen, strcat, strcpy, strcmp, strrev).

**Code:**

#include <stdio.h>

int main(){

int ch;

char s1[50],s2[50],c;

do{

system("cls");

printf ("\nProgram to implement major functions of string.h");

printf("\n1. Strlen\n2. Strcat\n3. Strcpy\n4. Strcmp\n5. Strrev");

printf("\nEnter your choice:- ");

scanf("%d",&ch); fflush(stdin);

switch(ch){

case 1:printf("Enter a string (max length - 50) : ");

gets(s1);

printf("Length of string '%s' is %d",s1,strlen(s1));

break;

case 2:printf("Enter first string (max length - 50) : ");

gets(s1);

printf("Enter second string (max length - 50) : ");

gets(s2);

printf("After concatenation, Result = %s",strcat(s1,s2));

break;

case 3:printf("Enter first string (max length - 50) : ");

gets(s1);

printf("Enter second string (max length - 50) : ");

gets(s2);

printf("After copying second string into first:-");

strcpy(s1,s2);

printf("\nFirst String = %s",s1);

printf("\nSecond String = %s",s2);

break;

case 4:printf("Enter first string (max length - 50) : ");

gets(s1);

printf("Enter second string (max length - 50) : ");

gets(s2);

if (strcmp(s1,s2)==0){

printf("\nBoth strings are equal");}

else{

break;

printf("\nBoth strings are not equal");}

case 5:printf("Enter a string (max length - 50) : ");

gets(s1);

char s[50];

strcpy(s,s1);

printf("Reverse of string '%s' is %s",s,strrev(s1));

break;

default:printf("\n Wrong choice");

break; }

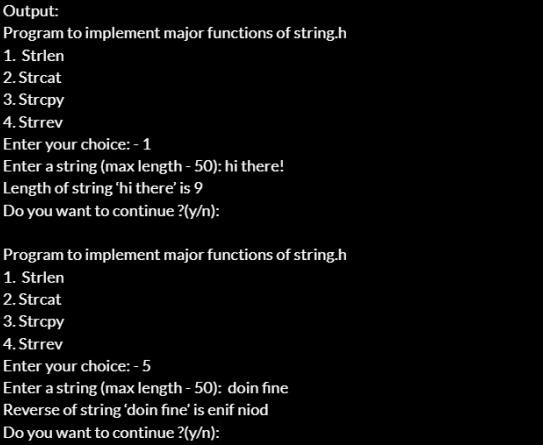
printf("\nDo you want to continue ? (y/n) : ");

scanf("%c",&c);

} while((c=='y')||(c=='Y'));

return 0; }

**Output:**



**VIVA QUESTIONS**

**Q. What is the use of string.h header while and where is this file stored?**

* String.h is the header file required for string functions. This function appends not more than n characters from the string pointed to by src to the end of the string pointed to by dest plus a terminating Null-character. It is stored in “/usr/include”.

**Q. How can we create a header file?**

* To make a header file, we have to create one file with a name, and extension should be (\*.h). In that function there will be no main() function. In that file, we can put some variables, some functions etc. To use that header file, it should be present at the same directory, where the program is located. Now using #include we have to put the header file name. The name will be inside double quotes. Include syntax will be look like this. #include ”header\_file.h”.

**Q. Write the function to find the length of a string.**

* int StrLength(char[] str){

int i;

for(i = 0; Str[i] != '\0'; ++i);

return i; }

**Q. Write the function to concatenate two strings.**

* #include <stdio.h> int main() {

char s1[100] = "programming ",

s2[] = "is awesome";

int length, j;

length = 0;

while (s1[length] != '\0') { ++length; } // concatenate s2 to s1

for (j = 0; s2[j] != '\0'; ++j, ++length) { s1[length] = s2[j]; }

s1[length] = '\0';

printf("After concatenation: ");

puts(s1);

return 0; }

**Q. Explain the use of header file.**

* A header file is a file with extension .h which contains C function declarations and macro definitions to be shared between several source files. There are two types of header files: the files that the programmer writes and the files that comes with your compiler.

**Experiment-2**

**Aim:** Write a program in C to reverse a linked list iterative and recursive.

**Code:**

#include <stdio.h>

#include <stdlib.h>

typedef struct node{

int i;

struct node \*next; } node1;

void reverselinkedlist(node1 \*head){

if(head->next==NULL)

printf("%d->",head->i);

else{

reverselinkedlist(head->next);

printf("%d->",head->i); }}

int main(){

char hh;

do{

char ch;

node1 \*head=NULL,\*tail=NULL;

do{

printf("Program to reverse a linked list (Iterative and Recursive)");

printf("\nCreating linked list...");

printf("\nEnter an integer:- ");

if(head==NULL){

head = malloc(sizeof(node1));

scanf("%d",&head->i);

head->next=NULL;

tail = head; }

else{

node1 \*n1 = malloc(sizeof(node1));

scanf("%d",&n1->i);

n1->next=NULL;

tail->next=n1;

tail=n1; }

printf("Linked list is:- ");

node1 \*n2=head;

printf(" ");

while(n2!=NULL){

printf("%d->",n2->i);

n2 = n2->next; }

printf("NULL");

printf("\nDo you want to enter more elements in the list (y/n) ? ");

fflush(stdin);

scanf("%c",&ch);

} while(ch=='y'||ch=='Y');

char ch1;

do{

int choice;

system("cls");

printf("Entered Linked List is:- ");

node1 \*n2=head;

printf(" ");

while(n2!=NULL){

printf("%d->",n2->i);

n2 = n2->next; }

printf("NULL");

printf("\n1. Reverse a linked list iteratively");

printf("\n2. Reverse a linked list recursively");

printf("\nEnter your choice - ");

scanf("%d",&choice);

switch(choice){

case 1: printf(" ");

node1 \*n2 = head;

node1 \*c,\*chead=NULL;

while(n2!=NULL){

c = malloc(sizeof(node1));

c->i = n2->i;

if(chead==NULL){

c->next = NULL;

chead = c; }

else{

c->next = chead;

chead = c;}

n2 = n2->next; }

node1 \*x = chead;

printf("\nReversed linked list is : ");

while(x!=NULL){

printf("%d->",x->i);

x=x->next; }

printf("NULL"); break;

case 2:printf("\nReversed linked list is : ");

reverselinkedlist(head);

printf("NULL"); break;

default:printf("\nWrong choice"); break; }

printf("Do you want to reverse linked list again using a different choice (y/n)? ");

fflush(stdin);

scanf("%c",&ch1);

} while((ch1=='Y')||(ch1=='y'));

printf("\nDo you want to run program again (y/n)? ");

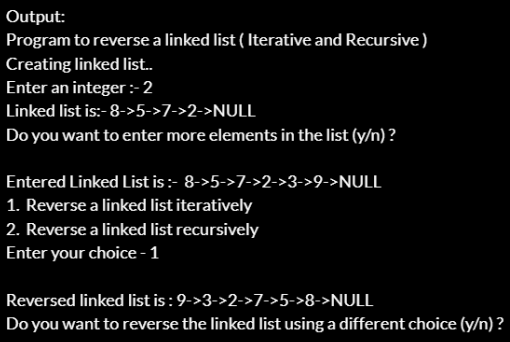
fflush(stdin);

scanf("%c",&hh);

} while((hh=='y')||(hh=='Y'));

return 0; }

**Output:**

****

**VIVA QUESTIONS**

**Q. What is the difference between iterative and recursive function call?**

* Recursion and iteration are both different ways to execute a set of instructions repeatedly. The main 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.

**Q. What are formal parameters in functions?**

* Formal parameters are always variables, while actual parameters do not have to be variables. Parameter Written in Function Call is Called “Actual Parameter”. One can use numbers, expressions, or even function calls as actual parameters.

**Q. How is the structure node declared?**

* typedef struct node{

int value;

struct node \*next;

} node;

node \*createNode(int val){

node \*newNode = malloc(sizeof(node));

newNode->value = val;

newNode->next = NULL;

return newNode; }

**Q. Define a link list.**

* A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers.

**Experiment-3**

**Aim:** Write a program in C to implement iterative Towers of Hanoi.

**Code:**

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <limits.h>

struct Stack {

unsigned capacity; int top;

int \*array; };

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; }

int isFull(struct Stack\* stack){

return (stack->top == stack->capacity - 1); }

int isEmpty(struct Stack\* stack) {

return (stack->top == -1); }

void push(struct Stack \*stack, int item) {

if (isFull(stack)) return;

stack -> array[++stack -> top] = item; }

int pop(struct Stack\* stack) {

if (isEmpty(stack))

return INT\_MIN;

return stack -> array[stack -> top--]; }

void moveDisk(char fromPeg, char toPeg, int disk) { printf("Move the disk %d from \'%c\' to \'%c\'\n", disk, fromPeg, toPeg); }

void moveDisksBetweenTwoPoles(struct Stack \*src, struct Stack \*dest, char s, char d) {

int pole1TopDisk = pop(src);

int pole2TopDisk = pop(dest);

if (pole1TopDisk == INT\_MIN) {

push(src, pole2TopDisk);

moveDisk(d, s, pole2TopDisk); }

else if (pole2TopDisk == INT\_MIN) {

push(dest, pole1TopDisk);

moveDisk(s, d, pole1TopDisk); }

else if (pole1TopDisk > pole2TopDisk) {

push(src, pole1TopDisk);

push(src, pole2TopDisk);

moveDisk(d, s, pole2TopDisk); }

else {

push(dest, pole2TopDisk);

push(dest, pole1TopDisk);

moveDisk(s, d, pole1TopDisk); }}

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 (num\_of\_disks % 2 == 0){

char temp = d; d = a; a = temp; }

total\_num\_of\_moves = pow(2, num\_of\_disks) - 1;

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);

}}

int main(){

unsigned num\_of\_disks = 3;

printf("Program Made By: Ayush Pandey\n");

struct Stack \*src, \*dest, \*aux;

src = createStack(num\_of\_disks);

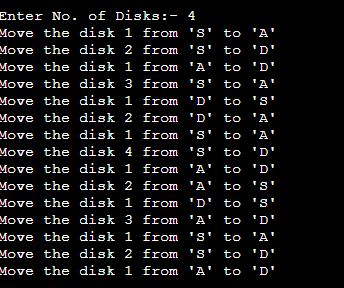
aux = createStack(num\_of\_disks);

dest = createStack(num\_of\_disks);

tohIterative(num\_of\_disks, src, aux, dest);

return 0; }

**Output:**

****

**VIVA QUESTIONS**

**Q. 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 following simple rules: Only one disk can be moved at a time.

**Q. What are the various ways is stack created?**

* There are two ways to implement a stack: Using array. Using linked list.

**Q. List the models of computation of language.**

* Models of computation can be classified into three categories: sequential models, functional models, and concurrent models.

**Q. What are objectives of principle of programming language?**

* Objectives are-

• To introduce several different paradigms of programming

• To gain experience with these paradigms by using example programming languages

• To understand concepts of syntax, translation, abstraction, and implementation

**Q. What are the Paradigms of Programming?**

* Paradigm can also be termed as method to solve some problem or do some task. Programming paradigm is an approach to solve problem using some programming language or also we can say it is a method to solve a problem using tools and techniques that are available to us following some approach. There are lots for programming language that are known but all of them need to follow some strategy when they are implemented and this methodology/strategy is paradigms. Apart from varieties of programming language there are lots of paradigms to fulfil each and every demand.

**Experiment-4**

**Aim:** WAP in C++ to count the numbers of object of a class with the help of static data member, function and constructor.

**Code:**

#include <iostream>

using namespace std;

class test{

int objNo;

static int objCnt;

public:

test(){

objNo = ++objCnt;

}

~test(){

--objCnt; }

void printObjNumber(void){

cout << "object number :" << objNo << "\n";}

static void printObjCount(void){

cout << "count:" << objCnt<< "\n";

}};

int test::objCnt;

int main(){

test t1, t2;

test::printObjCount();

test t3;

test::printObjCount();

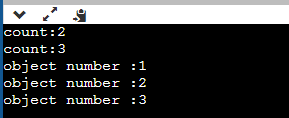
t1.printObjNumber();

t2.printObjNumber();

t3.printObjNumber();

return 0; }

**Output:**

****

**VIVA QUESTIONS**

**Q. List various type of languages.**

* 1. Procedural Programming Language

2. Object Oriented Programming Language

3. Parallel Processing Programming Language

4. Logic Programming Language

5. Functional Programming Language

6. Database Programming Language

**Q. What are the issues for languages?**

* 1. Development on hardware technologies.

2. Simplicity and performance trade off.

3. Usage Area

4. Portability

5. Variety of Project’s Design Patterns

6. Safety and security

**Q. What is translation?**

* Translation is the process of conversion of source code into the low level machine code.

**Q. What are different types of translation and their roles?**

* Compiler

A compiler is a translator used to convert high-level programming language to low-level programming language. It converts the whole program in one session and reports errors detected after the conversion.

* Interpreter

Just like a compiler, is a translator used to convert high-level programming language to low-level programming language. It converts the program one at a time and reports errors detected at once while doing the conversion.

* Assembler

An assembler is is a translator used to translate assembly language to machine language. It is like a compiler for the assembly language but interactive like an interpreter.

**Q. What is trade’s off of translation?**

* Trade’s off of translation are-

• Compilation – lower level machine may be faster, so programs run faster – compilation can be expensive – examples: C

• Interpretation – more ability to perform diagnostics (or changes) at run-time – examples: Basic, UNIX shells, Lisp

**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:**

1. **C++**

#include <iostream>

using namespace std;

class Time{

private: int hh;

int mm;

int ss;

public: Time(int h =0,int m = 0, int s = 0){

hh = h;

mm = m;

ss = s; }

Time add(Time t1){

Time temp;

temp.ss = ss + t1.ss;

temp.mm = mm + t1.mm;

temp.hh = hh + t1.hh;

if(temp.ss >= 60){

temp.mm += temp.ss / 60;

temp.ss = temp.ss % 60; }

if(temp.mm >= 60){

temp.hh += temp.mm / 60;

temp.mm = temp.mm % 60; }

return temp;}

void display(){

cout<<"HH:MM:SS = "<<hh<<":"<<mm<<":"<<ss<<endl; }};

int main(){

Time t1(4,35,55), t2(5, 10, 15), t3;

cout<<"Two times are:"<<endl;

t1.display();

t2.display();

t3 = t1.add(t2);

cout<<"SUM is:"<<endl;

t3.display();

return 0; }

1. **Java**

import java.util.\*;

public class Time {

public int hh, mm, ss;

public Time() {}

public Time(int hh, int mm, int ss) {

this.hh = hh;

this.mm = mm;

this.ss = ss; }

public void setTime(Time obj1, Time obj2) {

this.ss = obj1.ss + obj2.ss;

int extra = this.ss / 60;

this.ss %= 60;

this.mm = obj1.mm + obj2.mm + extra;

extra = this.mm / 60;

this.mm %= 60;

this.hh = obj1.hh + obj2.hh + extra;

this.hh %= 24; }

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int hh, mm, ss;

System.out.println("\n\t Time format --> HH:MM:SS");

System.out.print ("\t Enter Time 1:- ");

hh = scan.nextInt();

mm = scan.nextInt();

ss = scan.nextInt();

Time obj1 = new Time(hh, mm, ss);

System.out.print ("\t Enter Time 2:- ");

hh = scan.nextInt();

mm = scan.nextInt();

ss = scan.nextInt();

Time obj2 = new Time(hh, mm, ss);

Time obj3 = new Time();

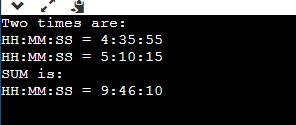
obj3.setTime(obj1, obj2);

System.out.print ("\t Sum = ");

System.out.print (obj3.hh + ":" + obj3.mm + ":" + obj3.ss);

} }

**Output:**

****

**VIVA QUESTIONS**

**Q. Write any four important uses of programming languages**

* 1. Machine Learning & Artificial Intelligence

2. Web and App Development

3. Data Analysis

4. Operating Systems & System Tools

**Q. 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 programs with the help of lexical symbols like its alphabet.

**Q. List the design principle of imperative languages.**

* It features close relation to machine architecture. It is based on Von Neumann architecture. It works by changing the program state through assignment statements. It performs step by step task by changing state. The main focus is on how to achieve the goal. The paradigm consist of several statements and after execution of all the result is stored.

**Q. Write the differences between array and enumerated data types in imperative languages?**

* Enums most often contain a group of related constants where the value of its first member is 0 and increases by one for each constant in the enum (by default). You can assign a value to an enum member and the value of the next member will be one higher.

An array is a block of contiguous memory that can be of any variable type: char, int, float, etc and its values can be changed throughout your program. If uninitialized the array will contain garbage values, unlike the enum.

**Q. Distinguish between dangling pointers and memory leakage.**

* Dangling pointer and memory leak are different terms. If a pointer is pointing to memory that is not owned by your program (except the null pointer) or an invalid memory, the pointer is called a dangling pointer. 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 leak occurs when you forget to deallocate the allocated memory. In the C language compiler does not deallocate the memory automatically it is freed by the programmer explicitly. In another word, you can say that a memory leak is a type of resource leak.

**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 and overload ‘\*’ operator to multiply two complex numbers using friend function of the class complex.

**Code:**

#include <iostream>

using namespace std;

class Complex{

private: float real;

float imag;

public:

Complex(): real(0), imag(0){ }

void input(){ int a[2];

for(int i=0; i<2; i++){ cin>>a[i]; }

real = a[0]; imag = a[1]; }

Complex operator + (Complex c){

Complex temp;

temp.real = real + c.real;

temp.imag = imag + c.imag;

return temp; }

friend Complex operator \* (Complex c1, Complex c2);

void output(){

if(imag < 0) cout << "Complex number: "<< real<< "i" << imag ;

else cout << "Complex number: " <<real << " + i" <<imag; }};

Complex operator \* (Complex c1, Complex c2) {

Complex c3;

c3.real=c1.real\*c2.real-c1.imag\*c2.imag;

c3.imag=c1.imag\*c2.imag;

return(c3); }

int main(){

Complex b, c, result;

int ch;

cout<<"\nEnter 1st complex number:- ";b.input();

cout<<"Enter 2nd complex number:- ";c.input();

cout<<"Complex Numbers operations\n1. Addition\n2. Multiplication";

cout<<"\nEnter choice:- ";cin>>ch;

switch(ch){

case 1:result = b + c;

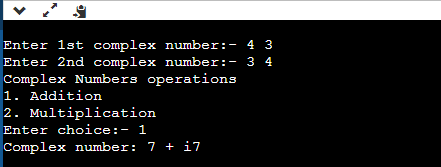
result.output(); break;

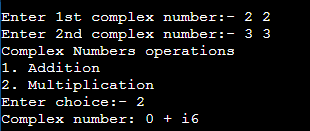
case 2:result = b\*c;

result.output(); break; }

return 0; }

**Output:**





**VIVA QUESTIONS**

**Q. List the benefits of modular development approach.**

* 1. Consistency

2. Reduced Development times

3. Flexibility

**Q. Give some reasons why computer scientists and professional software developers should study general concepts of language design and evaluation.**

* A solid programming languages foundation enables students to effectively recognize when designing a new language is appropriate and how to avoid these problems. Understanding programming language principles and models often provides the in- sights leading to new innovations as well.

**Q. What constitutes a programming environment?**

* File system, Text Editor, Linker, Compiler, Integrated tools.

**Q. Give an example of how aliasing deters reliability.**

* A pointer is a value that represents a memory address sometimes 2 pointers can represent the same memory address thats what aliasing is

int \* p;

\*p = 5;

int \* alias;

alias = p;

the variable alias is an alias of p and \*alias is equal to 5 if you change \*alias then \*p changes along with it.

**Q. How do type declaration statements effect the readability of programming language?**

* 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 it easy 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 mathematical operations parallel in Java.

**Code:**

**Server Class**

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 Class**

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);

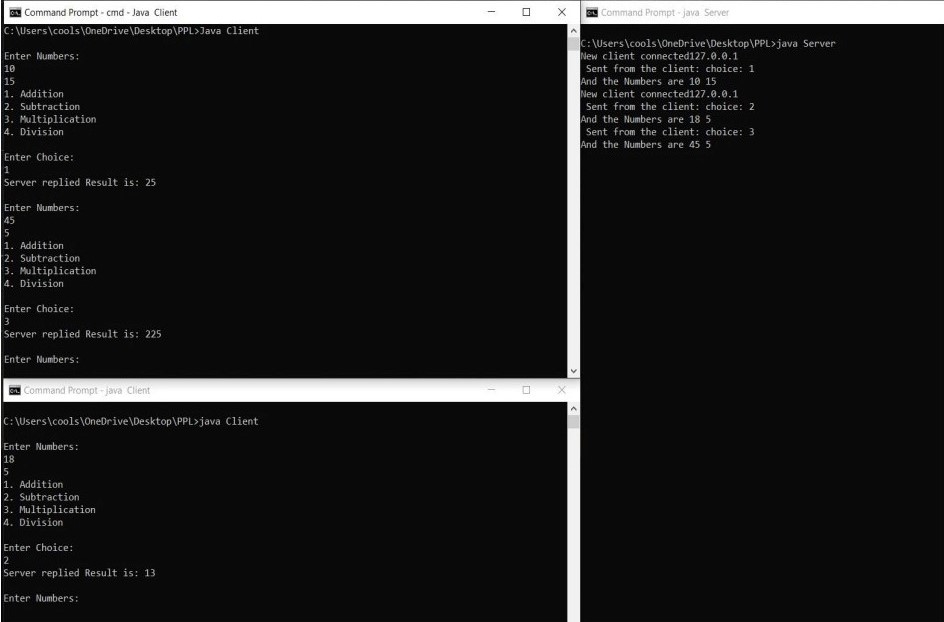
out.flush();

System.out.println("Server replied Result is: " + in.readLine()); }

sc.close(); }

catch (IOException e) { e.printStackTrace(); }}}

**Output:**

****

**VIVA QUESTIONS**

**Q. Write the uses of constructor and destructors in OOP.**

* Constructor is used to initialize an object of the class and assign values to data members 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.

**Q. Describe any one method for bridging the gap between high-level language and machine language**

* The compilation is a method whereby the source code is converted into object code. It is achieved with compiler assistance. The compiler tests the source code for syntactic or structural errors and produces the object code if the source code is error-free.

**Q. Explain language evaluation criteria and the characteristics that affect them.**

* Language Evaluation Criteria – Readability, Writability, Reliability, Cost.
* Characteristics that affect them are - Simplicity. Orthogonality. Data types and structures. Syntax. Support for Abstraction.
* Type checking. Exception handling. Aliasing. Control structures. Expressivity.

**Q. What Is Backus-naur Form (bnf)?**

* In computer science, Backus–Naur form (/ˌbækəs ˈnaʊər/) or Backus normal form (BNF) is a metasyntax notation 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.

**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();

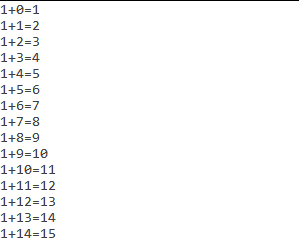
ans[i]=s2.str();

}

for(int i=0;i<50;i++) cout<<ques[i]<<"="<<ans[i]<<endl;

return 0; }

**Output:**

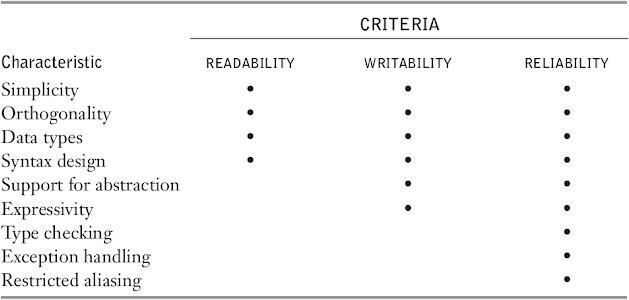




**VIVA QUESTIONS**

**Q. Explain language evaluation criteria and the characteristics that affect them.**

* The most prominent language evaluation criteria are:
* Readability
* Writability
* Reliability



**Q. 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.

**Q. 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 answerof these questions to the user now).

**Code:**

#include <iostream>

#include <string>

#include <sstream>

#include <ctime>

#include <stdlib.h>

using namespace std;

voidgenerateSetOfNumbers(intarr[]);

int main(){

stringques[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;}

voidgenerateSetOfNumbers(intarr[]) {

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];

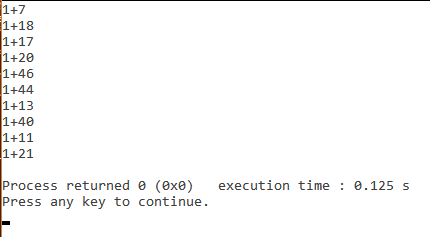
p[i] = p[j];

p[j] = temp; }

for (int i=0; i<10; ++i)

arr[i] = p[i]; }

**Output:**



**VIVA QUESTIONS**

**Q. List various types of pointers.**

* NULL pointer Dangling pointer
* Generic pointer Wild pointer
* Complex pointer

**Q. 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.

**Q. 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:**

package javaapplication4;

importjava.util.LinkedList;

public class JavaApplication4 {

public static void main(String[] args) throws InterruptedException {

final PC pc = new PC();

Thread t1 = new Thread(new Runnable(){

public void run(){ try { pc.produce(); }

catch(InterruptedException e) { e.printStackTrace(); }}});

Thread t2 = new Thread(new Runnable() {

public void run() { try { pc.consume(); }

catch(InterruptedException e) { e.printStackTrace(); }}});

t1.start();

t2.start();

t1.join();

t2.join(); }

public static class PC {

LinkedList<Integer> list = new LinkedList<>();

int capacity = 2;

public void produce() throws InterruptedException {

int value = 0;

while (true) {

synchronized (this) {

while (list.size()==capacity)

wait();

System.out.println("Producer produced-" + value); list.add(value++);

notify();

Thread.sleep(1000); }}}

public void consume() throws InterruptedException{

while (true) {

synchronized (this) {

while (list.size()==0)

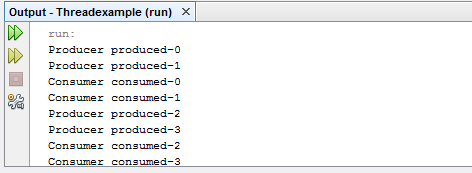
wait();

intval = list.removeFirst(); System.out.println("Consumer consumed-"+ val);

notify();

Thread.sleep(1000); }}}}}

**Output:**



**VIVA QUESTIONS**

**Q. 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.

**Q. 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.

**Q. How are threads created?**

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

**Experiment-11**

**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;

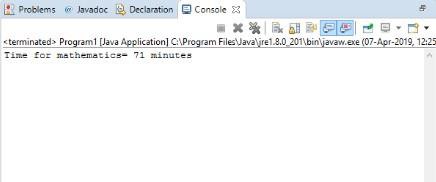
int mathsQues= 50, ques= 150, t;

t= time/(ques + 2\*mathsQues);

System.out.println("Time for mathematics= "+(mathsQues\*2\*t)/60 + "minutes");

}}

**Output:**



**Experiment-12**

**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:**

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); }

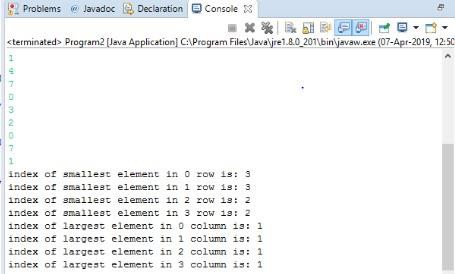
for(int j=0;j<m;j++){

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:**



**Experiment-13**

**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);

}}

**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[]);

void 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);}

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++)

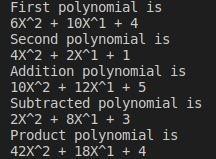
for(j=0;j<3;j++)

p3[i+j]=p3[i+j]+p1[i]\*p2[j];

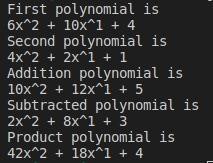
print(p3); }

**Output:**

**Java**



**C++**



**Experiment-14**

**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;

}} }

**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;

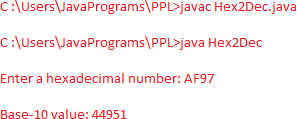
case 'f':

case 'F': return 15;

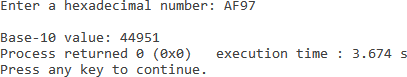
default: return -1; }}

**Output:**

**Java**



**C++**



**Experiment-15**

**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){

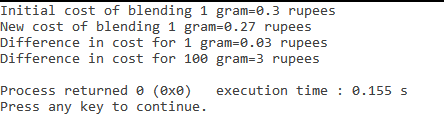
int costOfNGram=x\*33+y\*24;

int costOfOneGram=(x\*33+y\*24)/(x+y);

return costOfOneGram;

}

**Output:**



**Experiment-16**

**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;

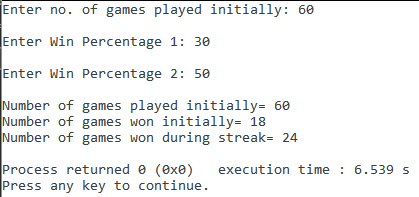
gamesWon1 = (gamesPlayed1\*winPercentage1)/100;

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

return gamesWon2;

}

**Output:**



**Experiment-17**

**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; if (b == 0) return a; if (a == b)

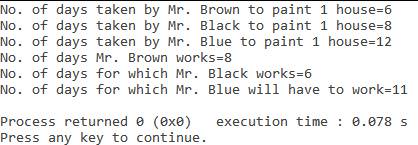
return a; if (a > b)

return gcd(a-b, b);

return gcd(a, b-a);

}

**Output:**



**Experiment-18**

**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:**

