|  |  |  |
| --- | --- | --- |
| COMPLETE C FILE    Here I have tried to touch almost every topic in C programming language.  Here you can learn the complex file handling concepts very easily, how to use pointer actually, dynamic memory allocation and more… |  | C programs… |

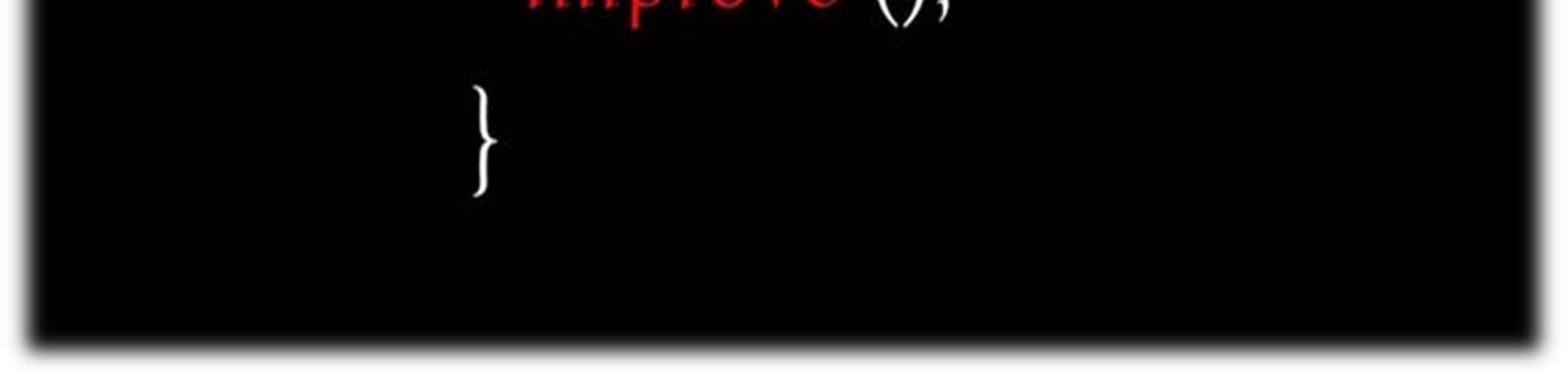
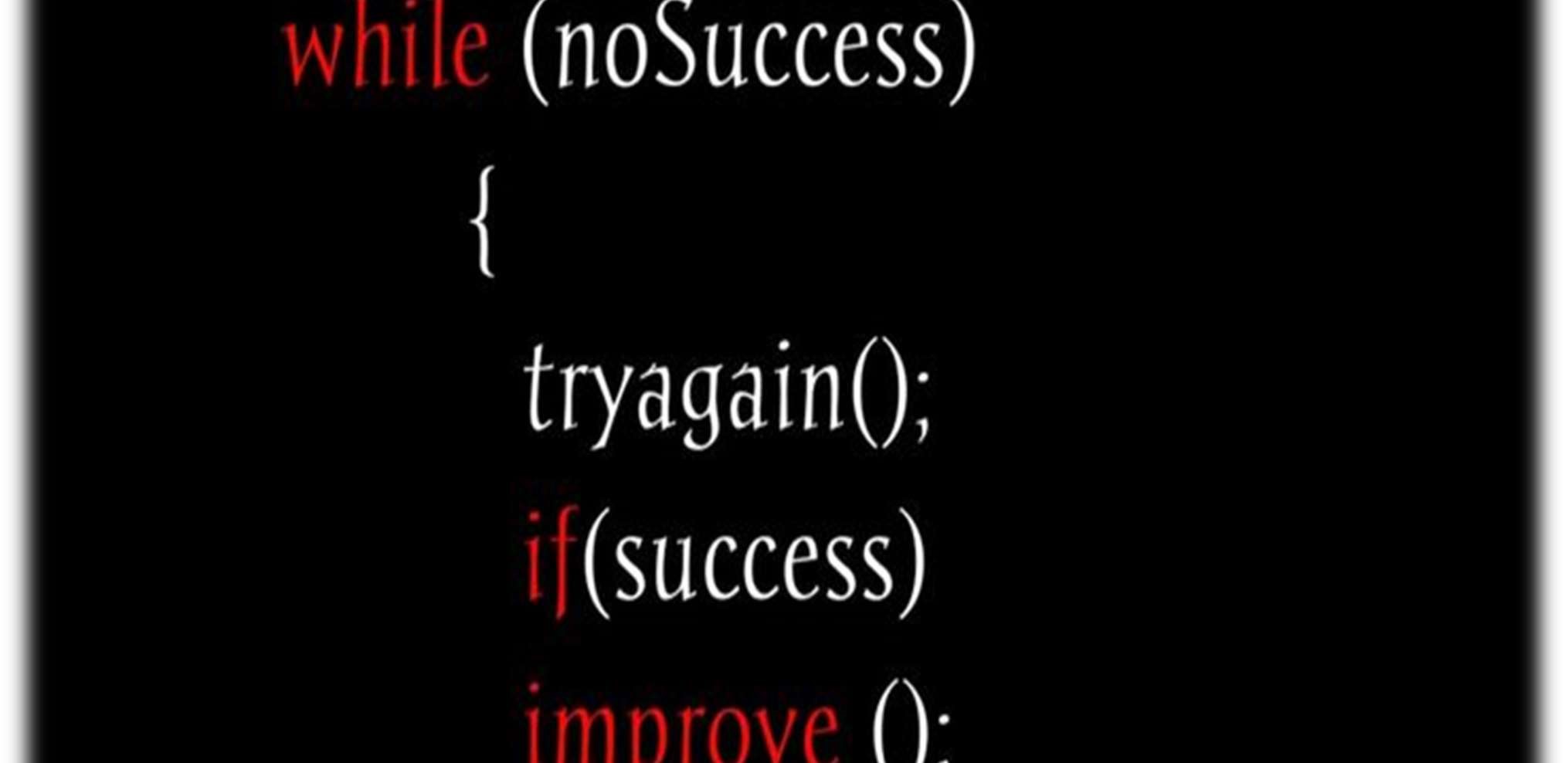
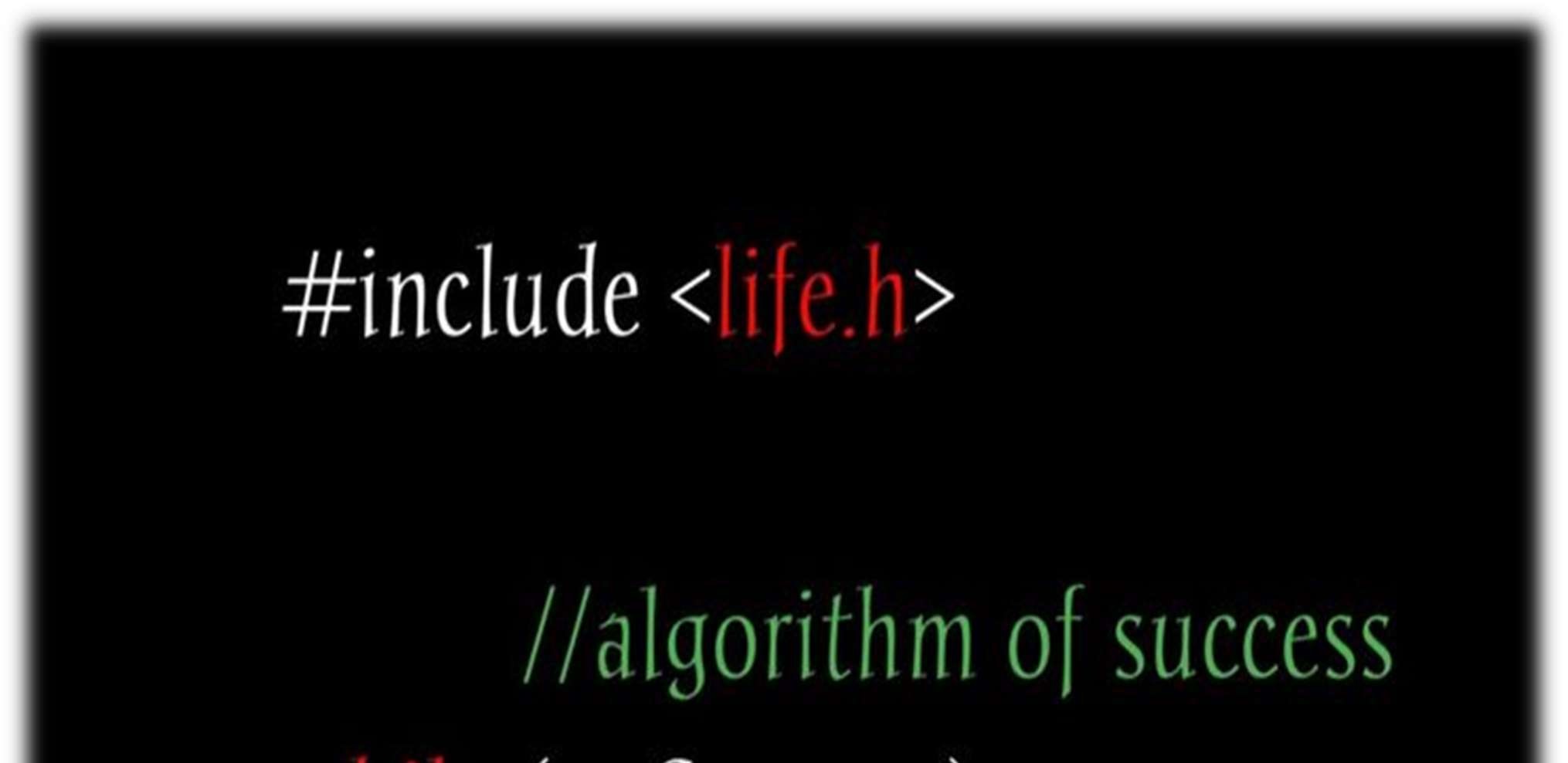


Table of content

Practical 1: Program to implement conditional statements in C language.

Write following program for conditional statements:

1. Write a program in C to enter two number and find out greater one.
2. Write a program in C to enter 3 number and find out greater one. Practical 2: Program to implement switch-case statement in C language

Write following programs for switch-case:

1. Write a program in C to enter day of week and display day name of that week.
2. Write a program in C to enter month number and display month name.

Practical 3: Program to implement looping constructs in C language.

Write following program for looping:

1. Write a program in C to display number 1 to 10 using for loop, while loop and do...while loop.
2. Write a program in C to get sum of number 1 to 100 using any loop
3. Write a program in C to make small calculator that ask to enter choice for various operations addition, subtraction, multiplication and division on numbers using do...while loop.
4. Write a program in C to display table of number 1 to 10 using nested loop.

Practical 4: Program to perform basic input-output operations in C language.

Write following programs for input-output concepts:

1. Write program in C to get input by using gets() function.
2. Write program in C to display information using puts() function.
3. Write program in C that show difference between getch() and getche() functions.
4. Write program in C to get input from keyboard using scanf() functions and perform some operations on that input.

Practical 5: Program to implement user defined functions in C language.

Write following programs for functions.

1. Write program in C to make small calculator using functions.
2. Write program in C to get greater number among 3 numbers using functions.
3. Write program in C to pass your name as parameter in functions and that function should display "Welcome to C Function <Your Name Here>".
4. Write program in C to get factorial of a number.
5. Write program in C to get Fibonacci Series using function.

Practical 6: Program to implement recursive functions in C language.

Write a following program for recursion:

1. Write a program in C to display 1 to number you get enter by keyboard using recursion.
2. Write program in C to get factorial of a number using recursion.
3. Write program in C to get Fibonacci Series using recursion.
4. Write a program to implement Ackermann function using recursion
5. Write a program in C to enter a number and check it is Armstrong or not.

Practical 7: Program to implement one-dimensional arrays in C language.

Write a following program for 1D Arrays:

* 1. Write a program in C to enter 10 values in array and display using for loop.
  2. Write a program in C to enter values in array and display reverse of it.
  3. Write a program in C to delete number from array at given position.
  4. Write a program in C to insert number in array at specific position.
  5. Write a program in C to sort array using Bubble Sort.
  6. Write a program in C to search a value in Array.

Practical 8: Program to implement two-dimensional arrays in C language

Write a following program for 2D Arrays:

* 1. Write a program in C to enter some value in 2D array and display it using any loop
  2. Write a program in C to perform addition of matrix on 2D array.
  3. Write a program in C to perform multiplication of matrix on 2D array.

Practical 9: Program to implement or use pointers concepts in C language.

Write following programs for pointers:

* 1. Write a C program to create, initialize and use pointers.
  2. Write a C program to add two numbers using pointers.
  3. Write a C program to swap two numbers using pointers.
  4. Write a C program to input and print array elements using pointer.
  5. Write a C program to copy one array to another using pointers.
  6. Write a C program to swap two arrays using pointers.
  7. Write a C program to reverse an array using pointers.
  8. Write a C program to search an element in array using pointers.
  9. What will be the output of the C program?

#include<stdio.h> void function(char\*\*); int main() {

char \*arr[] = { "ant", "bat", "cat", "dog", "egg", "fly" }; function(arr); return 0; } void function(char \*\*ptr)

{ char \*ptr1;

ptr1 = (ptr += sizeof(int))[-2]; printf("%s\n", ptr1);

}

Practical 10: : Program to implement or use double pointers and array of pointers concepts in C language.

Write following programs for pointers:

1. Write C program to demonstrate pointer to pointer or double pointer.
2. Write C program to demonstrate array using pointer to pointer or double pointer.
3. Write a program to add two 2 X 2 matrix using pointers.
4. Write a program to multiply two 2 X 2 matrix using pointers.

Practical 11: Program to implement string manipulation functions in C language Write following programs for strings:

1. Write C program to demonstrate string using pointer to pointer or double pointer.
2. Write a program to find length of string using string array.
3. Write a program to copy one string to another using string array.
4. Write a program to concatenate two strings using string array.
5. Write a program to compare two strings using string array.
6. Write a program to find length of string using pointers.
7. Write a program to copy one string to another using pointer.
8. Write a program to concatenate two strings using pointers.
9. Write a program to compare two strings using pointers.

Practical 12: Program to implement structure and union in C language.

Write following programs for structure and union:

1. Write C program to create, declare and initialize structure.
2. Write C program to read and print an employee's detail using structure.
3. Write C program to demonstrate example of nested structure.
4. Write C program to demonstrate example structure pointer (structure with pointer).
5. Write C program to declare, initialize a union, example of union.
6. Write C program to demonstrate example of structure of array.
7. Write C program to show different between structure and union.
8. Write C program to use typedef for creating structure and union.

Practical 13: Program to implement macro and storage classes in C language

Write following programs for strings:

1. Write C program to define the math operator ‘+’ as PLUS, ‘-‘ as MINUS, ‘\*’ as MULT & ‘/’ as DIVIDE using pre-processor directives and do the operations over variables (x, y) defined on above question like z=x PLUS y.
2. Write C program to define a macro with one parameter to compute the volume of a sphere. Write a C program using this macro to compute the volume for spheres of radius 5, 10 and 15 meters.
3. Write C program to define a macro that receives an array and the number of elements in the array as arguments. Write a C program for using this macro to print the elements of the array.
4. Write C program to illustrate the properties of a static variable.
5. Write C program to illustrate the properties of an auto variable.
6. Write C program to illustrate the properties of an extern variable. 7. Write C program to illustrate the properties of a register variable.

Practical 14: Program to perform operations using malloc, calloc, realloc, free and on file handling in C language.

Write following programs for strings:

1. Write C program to read file using fgetc(), fgets(), and fscanf() functions.
2. Write C program to write data into any file using fputc(), gputs(), and fprintf() functions.
3. Write C program to copy character or binary file.
4. Write C program to create dynamic array using malloc() function.
5. Write C program to create dynamic array using calloc() function.
6. Write C program to create dynamic array and re-size array using realloc() function.

Practical 15: Program to perform graphical operations in C language.

Write following programs for strings:

1. Write C program to draw line.
2. Write C program to draw circle.
3. Write C program to draw rectangle.
4. Write C program to move circle one location to another on pressing enter key.

Practical 1: Program to implement conditional statements in C language.

//Write a program in C to enter two number and find out greater one.

#include<stdio.h> #include<conio.h> void main() { int n1,n2; printf("Enter the first number: "); scanf("%d",&n1); printf("Enter the Second number: "); scanf("%d",&n2); if(n1>n2) printf("first number is greater"); else printf("\nSecond number is greater");

}

//Write a program in C to enter 3 number and find out greater one.

#include<stdio.h> #include<conio.h> void main() { int n1,n2,n3; printf("Enter the first number: "); scanf("%d",&n1); printf("Enter the Second number: "); scanf("%d",&n2); printf("Enter the Third number: "); scanf("%d",&n3); if(n1>n2 && n1>n3) printf("first number is greater"); else if(n2>n3) printf("\nSecond number is greater"); else printf("\nThird number s=is greater");

}

Practical 2: Program to implement switch-case statement in C language

//Write a program in C to enter day of week and display day name of that week

#include<stdio.h> #include<conio.h> void main() { int n; printf("Enter the day number: "); scanf("%d",&n); switch(n) { case 1: printf("Sunday"); break; case 2: printf("Monday"); break; case 3: printf("Tuesday"); break; case 4: printf("Wednesday"); break; case 5: printf("Thursday"); break; case 6: printf("Friday"); break; case 7: printf("Saturday"); break; default:

printf("Invalid input");

}

}

//Write a program in C to enter month number and display month name.

#include<stdio.h> #include<conio.h> void main() { int n; printf("Enter the month number: "); scanf("%d",&n); switch(n) { case 1: printf("J=January"); break; case 2: printf("February"); break; case 3: printf("March"); break; case 4: printf("April"); break; case 5: printf("May"); break; case 6: printf("June"); break; case 7: printf("July"); break; case 8: printf("August"); break; case 9: printf("September"); break; case 10: printf("October"); break;

case 11: printf("November"); break; case 12: printf("December"); break; default: printf("Invalid input");

}

}

Practical 3: Program to implement looping constructs in C language.

//Write a program in C to display number 1 to 10 using for loop, while loop and do...while loop. #include<stdio.h> #include<conio.h> void main() { int i; printf("Using for loop\n"); for(i=1;i<=10;i++)

{ printf("%d ",i);

} printf("\nUsing while loop\n"); i=1; while(i<=10) { printf("%d ",i); i++;

} printf("\nUsing do- while loop\n"); i=1; do{ printf("%d ",i); i++;

}while(i<=10);

}

//2. Write a program in C to get sum of number 1 to 100 using any loop

#include<stdio.h> #include<conio.h> void main() {

int i,sum=0;

for(i=1;i<=100;i++)

{ sum+=i; } printf("Sum from 1 to 100 is %d",sum);

}

//3. Write a program in C to make small calculator that ask to enter choice /// for various operations addition, subtraction, multiplication and division on numbers using do...while loop.

#include <stdio.h> #include <stdlib.h> int main() { int a, b, i; printf("Enter Number 1 :--> "); scanf("%d", &a); printf("\nEnter Number 2 :--> "); scanf("%d", &b); do {

printf("\nEnter 1 for '+' "); printf("\nEnter 2 for '-' "); printf("\nEnter 3 for '\*' "); printf("\nEnter 4 for '/' \n"); printf("Enter 0 for Exit "); printf("\nEnter your choice :--> "); scanf("%d", &i); switch(i) { case 0: break; case 1: printf("\na + b = %d\n", a + b); break; case 2: printf("\na - b = %d\n", a - b); break; case 3: printf("\na \* b = %d\n", a \* b); break; case 4: printf("\na / b = %f\n", a / b); break; default:

printf("\nEnter valid option\n");

}

} while(i != 0);

}

//4. Write a program in C to display table of number 1 to 10 using nested loop.

#include <stdio.h> #include <stdlib.h> int main() { int i,j;

for(i=1;i<=10;i++)

{ printf("-----------Table of %d---------\n",i); for(j=1;j<=10;j++) printf("%d x %d = %d\n",i,j,i\*j);

}

}

Practical 4: Program to perform basic input-output operations in C language.

//1. Write program in C to get input by using gets() function.

//2. Write program in C to display information using puts() function.

#include <stdio.h> #include <stdlib.h> int main() { char name[100]; printf("Enter your name pls..."); gets(name); printf("\nYou have entred... "); puts(name);

}

//3. Write program in C that show difference between getch() and getche() functions.

#include <stdio.h> #include <stdlib.h> int main() { char ch; printf("Enter the gender of employee(m/f)..."); ch=getch();

printf("\nContinue...press (y/n)"); ch=getche(); printf("You have choosen %c",ch);

}

//4. Write program in C to get input from keyboard using scanf() functions and perform some operations on that input.

#include <stdio.h> #include <stdlib.h> int main() { int a,b,s=0; printf("Enter the first number:"); scanf("%d",&a);

printf("Enter the second number:"); scanf("%d",&b); s=a+b;

printf("Sum is %d",s);

}

Practical 5: Program to implement user defined functions in C language.

//1. Write program in C to make small calculator using functions.

#include <stdio.h> int add(int a, int b)

{

return a+b;

} int mul(int a, int b)

{

return a\*b;

} int sub(int a, int b)

{

return a-b;

} int div(int a, int b)

{

return a/b;

} int rem(int a, int b)

{

return a%b;

} int main() { int a,b,res; char ch; printf("Enter the first number: "); scanf("%d",&a); printf("\nEnter the second number: "); scanf("%d",&b); fflush(stdin); printf("--------------MENU-----------------"); printf("\nChoose an operator you want:"); printf("\n+\n-\n\*\n/\n\%%\n\t: "); scanf("%c",&ch); switch(ch) { case '+':

res=add(a,b); printf("\nResult= %d",res); break; case '-':

res=sub(a,b); printf("\nResult= %d",res); break; case '/':

res=div(a,b); printf("\nResult= %d",res); break; case '\*':

res=mul(a,b); printf("\nResult= %d",res); break; case '%':

res=rem(a,b); printf("\nResult= %d",res); break; default: printf("\nINVAlID INPUT"); } return 0;

}

//2. Write program in C to get greater number among 3 numbers using functions.

#include<stdio.h>

int greater(int a,int b,int c)

{ if(a>b && a>c ) return a; else if(b>c) return b; else return c;

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

printf("Enter the 1st number: "); scanf("%d",&a); printf("\nEnter the 2nd number: "); scanf("%d",&b); printf("\nEnter the 3rd number: "); scanf("%d",&c); g=greater(a,b,c);

printf("\nGreatest is: %d",g);

}

//3. Write program in C to pass your name as parameter in functions and that function should display "Welcome to C Function <Your Name Here>"

#include<stdio.h> void fun(char c[50])

{ printf("Welcome to C Function %s",c);

} int main() { char name[50]; printf("Enter your name: "); gets(name); fun(name); return 0;

}

//4. Write program in C to get factorial of a number.

#include<stdio.h> int fact(int n)

{ if(n==1) return 1; else if(n==0) return 1; return n\*fact(n-1);

} int main() { int n; printf("Enter a number: "); scanf("%d",&n); printf("\nIt's factorial is : %d",fact(n)); return 0;

}

//5. Write program in C to get Fibonacci Series using function.

#include<stdio.h> void fibo(int a,int b,int n) //int fibo(firstTerm,SecondTerm,TotalTerm)

{ int c; if(n<=1) printf("Invalid argument!"); else{ printf("%d %d",a,b); n-=2; while(n>0) { c=a+b; a=b; b=c; printf(" %d",c); n--;

}

} } int main() { int a,b,n;

printf("Enter the first term: "); scanf("%d",&a); printf("\nEnter the second term: "); scanf("%d",&b);

printf("\nEnter the length of the fibonacci you want: "); scanf("%d",&n); fibo(a,b,n); return 0;

}

Practical 6: Program to implement recursive functions in C language.

//1. Write a program in C to display 1 to number you get enter by keyboard using recursion. #include<stdio.h> void display(int num)

{ if(num) display(num-1); else return;

printf("%d\n", num);

} int main() { int n; printf("Enter the number: "); scanf("%d",&n); display(n);

return 0;

}

//2. Write program in C to get factorial of a number using recursion.

#include<stdio.h> int fact(int n)

{ if(n==1) return 1; else if(n==0) return 1; return n\*fact(n-1);

} int main() { int n; printf("Enter a number: "); scanf("%d",&n); printf("\nIt's factorial is : %d",fact(n)); return 0;

}

//3. Write program in C to get Fibonacci Series using recursion.

#include<stdio.h> void fibo(int n){ static int n1=0,n2=1,n3; if(n>0){ n3 = n1 + n2; n1 = n2; n2 = n3; printf("%d ",n3); fibo(n-1);

}

} int main() { int n; printf("Enter the lenght for fibonacci: "); scanf("%d",&n); fibo(n); return 0;

}

//4. Write a program to implement Ackermann function using recursion

#include <stdio.h> int ack(int m, int n)

{ if (m == 0){ return n+1;

} else if((m > 0) && (n == 0)){ return ack(m-1, 1);

} else if((m > 0) && (n > 0)){ return ack(m-1, ack(m, n-1));

}

} int main(){ int n,m; printf("Enter the value of m: "); scanf("%d",&m); printf("Enter the value of n: "); scanf("%d",&n); printf("The result is: %d",ack(m,n)); return 0;

}

//5. Write a program in C to enter a number and check it is Armstrong or not.

#include<stdio.h>

#include<conio.h> #include<math.h> int is\_armstrong(int a)

{ int sum\_of\_digits(int); if(sum\_of\_digits(a)==a) return 1; else return 0;

} int power(int b,int p)

{ int i,res=1; for(i=1;i<=p;i++)

{

res=res\*b;

} return res;

} int sum\_of\_digits(int a)

{ int power(int ,int); int c=0,temp,sum=0; temp=a; while(a>0) { a=a/10; c++; } a=temp; while(a>0) { temp=a%10; sum=sum+power(temp,c); a=a/10; } return sum;

}

int is\_prime(int a)

{ int i; for(i=2;i<sqrt(a);i++)

{ if(a%i==0) return 0; } return 1; } void main() { int ch,i,num; int is\_armstrong(int); int is\_prime(int); printf("Enter a number:"); scanf("%d",&num); printf("\n-----------MENU-----------"); printf("\n1.Check for armstrong"); printf("\n2.Check for prime"); printf("\nEnter your choice:"); scanf("%d",&ch); switch(ch) { case 1: if(is\_armstrong(num))

{ printf("\nNumber is armstrong");

} else printf("\nNumber is not armstrong"); break; case 2: if(is\_prime(num))

{ printf("\nNumber is prime");

} else

printf("\nNumber is not prime"); break; default:

printf("\nWrong choice!");

}

}

Practical 7: Program to implement one-dimensional arrays in C language.

//1.Write a program in C to enter 10 values in array and display using for loop.

#include<stdio.h>

int main() { int a[10],i; printf("Enter 10 elements you want: "); for(i=0;i<10;i++) scanf("%d",&a[i]);

printf("\nYour array is as follows: "); for(i=0;i<10;i++) printf("%d\t",a[i]); return 0;

}

//2.Write a program in C to enter values in array and display reverse of it.

#include<stdio.h>

int main() { int a[10],i; printf("Enter 10 elements you want: "); for(i=0;i<10;i++) scanf("%d",&a[i]); printf("\nYour array in reverse order is: "); for(i=9;i>=0;i--) printf("%d\t",a[i]); return 0;

}

//3.Write a program in C to delete number from array at given position.

#include<stdio.h>

int main() { int a[10],i,pos; printf("Enter 10 elements you want: "); for(i=0;i<10;i++) scanf("%d",&a[i]); printf("\nEnter the position from which you want to delete: "); scanf("%d",&pos);

for(i=pos-1;i<9;i++) a[i]=a[i+1]; printf("\nUpdated array: "); for(i=0;i<9;i++) printf("%d ",a[i]); return 0;

}

//4. Write a program in C to insert number in array at specific position.

#include<stdio.h>

int main() { int a[12],i,n,pos; printf("Enter 10 elements you want: "); for(i=0;i<10;i++) scanf("%d",&a[i]); printf("Enter the number you want to insert: "); scanf("%d",&n); printf("\nEnter the position at which you want to insert: "); scanf("%d",&pos);

for(i=10;i>=pos;i--) a[i]=a[i-1]; a[i]=n; printf("\nUpdated array: "); for(i=0;i<11;i++) printf("%d ",a[i]); return 0;

}

//5. Write a program in C to sort array using Bubble Sort.

#include<stdio.h>

int main() { int a[12],i,t,j; printf("Enter 10 elements you want: "); for(i=0;i<10;i++) scanf("%d",&a[i]); for(i=0;i<10;i++)

{ for(j=0;j<10-i-1;j++)

{

if(a[j]>a[j+1])

{ t=a[j]; a[j]=a[j+1]; a[j+1]=t;

}

}

}

printf("\nUpdated array: "); for(i=0;i<10;i++) printf("%d ",a[i]); return 0;

}

//6. Write a program in C to search a value in Array.

#include<stdio.h> int main() { int a[12],i,t,pos=-1; printf("Enter 10 elements you want: "); for(i=0;i<10;i++) scanf("%d",&a[i]); printf("\nEnter the value you want to search: "); scanf("%d",&t); for(i=0;i<10;i++)

{

if(a[i]==t) { pos=i; break;

} } if(pos>=0) { printf("\nValue is present at %d index",pos);

} else printf("Value not found"); return 0;

}

Practical 8: Program to implement two-dimensional arrays in C language

//1. Write a program in C to enter some value in 2D array and display it using any loop #include<stdio.h> void display(int a[10][10],int r,int c)

{ int i,j; for(i=0;i<r;i++)

{ for(j=0;j<c;j++)

{ printf("%d ",a[i][j]);

}

printf("\n");

} } void input(int a[10][10],int r,int c)

{ int i,j; for(i=0;i<r;i++)

{ for(j=0;j<c;j++)

{ scanf("%d",&a[i][j]);

}

} } int main() { int a[10][10],b[10][10],x[10][10], i,j,r,c;

printf("Enter the number of rows: "); scanf("%d",&r); printf("\nEnter the number of coloumns: "); scanf("%d",&c);

printf("\nEnter %d values for 1st matrix: ",r\*c); input(a,r,c); printf("\n-------Your Matrix------------\n"); display(a,r,c); return 0;

}

//2. Write a program in C to perform addition of matrix on 2D array.

#include<stdio.h> void add\_matrix(int a[10][10],int b[10][10],int x[10][10], int r,int c)

{ int i,j; for(i=0;i<r;i++)

{ for(j=0;j<c;j++)

x[i][j]=a[i][j]+b[i][j];

} } void display(int a[10][10],int r,int c)

{ int i,j; for(i=0;i<r;i++)

{ for(j=0;j<c;j++)

{ printf("%d ",a[i][j]);

}

printf("\n");

} } void input(int a[10][10],int r,int c)

{ int i,j; for(i=0;i<r;i++)

{ for(j=0;j<c;j++)

{ scanf("%d",&a[i][j]);

}

} } int main() {

int a[10][10],b[10][10],x[10][10], i,j,r,c;

printf("Enter the number of rows: "); scanf("%d",&r);

printf("\nEnter the number of coloumns: "); scanf("%d",&c); printf("\nEnter %d values for 1st matrix: ",r\*c); input(a,r,c);

/\*since both the matrix should me equivalent for addition

therefor I'm using the same row and column for 2nd matrix as well\*/ printf("\nEnter %d values for 2st matrix: ",r\*c); input(b,r,c); add\_matrix(a,b,x,r,c); //calling additoin function for x=a+b printf("\n-------1st matrix------------\n"); display(a,r,c); printf("\n-------2nd matrix-------------\n"); display(b,r,c); printf("\n-------resultant matrix------------\n"); display(x,r,c); return 0;

}

//3. Write a program in C to perform multiplication of matrix on 2D array.

#include<stdio.h> void mul\_matrix(int a[10][10],int b[10][10],int x[10][10], int r1,int c2,int r2)

{ int i,j,k; for(i=0;i<r1;i++)

{ for(j=0;j<c2;j++)

{ x[i][j]=0; for(k=0;k<r2;k++)

{ x[i][j]+=a[i][k]\*b[k][j];

}

}

} } void display(int a[10][10],int r,int c)

{ int i,j; for(i=0;i<r;i++)

{ for(j=0;j<c;j++)

{ printf("%d ",a[i][j]);

}

printf("\n");

} }

void input(int a[10][10],int r,int c)

{ int i,j; for(i=0;i<r;i++)

{ for(j=0;j<c;j++)

{

scanf("%d",&a[i][j]);

}

} } int main() { int a[10][10],b[10][10],x[10][10], i,j,r1,c1,r2,c2; printf("Enter the number of rows for 1st matrix: "); scanf("%d",&r1); printf("\nEnter the number of coloumns for 1st matrix: "); scanf("%d",&c1);

printf("Enter the number of rows for 2nd matrix: "); scanf("%d",&r2); printf("\nEnter the number of coloumns for 2nd matrix: "); scanf("%d",&c2); if(c1!=r2) { printf("Multiplication is not possible since c1 is not equla to r2"); return 0;

}

printf("\nEnter %d values for 1st matrix: ",r1\*c1); input(a,r1,c1); printf("\nEnter %d values for 2st matrix: ",r2\*c2); input(b,r2,c2); mul\_matrix(a,b,x,r1,c2,r2); //calling multiplication function for x=a\*b printf("\n-------1st matrix------------\n"); display(a,r1,c1); printf("\n-------2nd matrix-------------\n"); display(b,r2,c2); printf("\n-------resultant matrix------------\n"); display(x,r1,c2); return 0;

}

Practical 9: Program to implement or use pointers concepts in C language.

//1. Write a C program to create, initialize and use pointers.

//2. Write a C program to add two numbers using pointers.

#include<stdio.h>

int main() { int a=5,b=3,sum; int \*p,\*q; //Declaration of pointers p=&a; //initializing with address of a q=&b; //initializing with address of b sum=\*p+\*q; // It behaves like sum = a+b printf("Sum= %d",sum); return 0;

}

//3. Write a C program to swap two numbers using pointers.

#include<stdio.h> void swap(int \*a,int \*b)

{ int t; t=\*a; \*a=\*b;

\*b=t; } int main() { int a,b;

printf("Enter the value for a: "); scanf("%d",&a);

printf("Enter the value for b: "); scanf("%d",&b); printf("\n--------Before swapping--------\na=%d\nb=%d",a,b); swap(&a,&b); printf("\n--------After swapping--------\na=%d\nb=%d",a,b);

return 0;

}

//4. Write a C program to input and print array elements using pointer.

#include<stdio.h> int main() { int a[10],i; int \*p; p=&a; printf("Enter the 10 elements: "); for(i=0;i<10;i++) scanf("%d",p+i); //input using pointer printf("You have entered: "); for(i=0;i<10;i++)

{ printf("%d ",\*(p+i)); //displaying value using pointer

}

return 0;

}

//5. Write a C program to copy one array to another using pointers.

#include<stdio.h> int main() { int a[10],i,b[10]; int \*p,\*q; p=a; q=b; printf("Enter the 10 elements: "); for(i=0;i<10;i++) scanf("%d",p+i); //input using pointer for(i=0;i<10;i++)

{

\*(q+i)=\*(p+i); //It will behave like b[i]=a[i]

} printf("\nArray b is: "); for(i=0;i<10;i++)

{ printf("%d ",\*(q+i));

} return 0;

}

//6. Write a C program to swap two arrays using pointers.

#include<stdio.h> void disp(int \*a)

{ for(int i=0;i<5;i++) printf("%d ",\*(a+i));

} int main() { int a[10],i,b[10]; int \*p,\*q,t; p=a; q=b; printf("Enter the 5 elements for 1st array: "); for(i=0;i<5;i++) scanf("%d",p+i);

printf("Enter the 5 elements for 2nd array: "); //Since for swapping both the variable should have the same space for(i=0;i<5;i++) scanf("%d",q+i);

printf("\n----------BEFORE SWAPPING------------\n "); printf("a=");disp(p); printf("\nb=");disp(q); for(i=0;i<5;i++)

{

t=\*(q+i);

\*(q+i)=\*(p+i);

\*(p+i)=t;

}

printf("\n----------AFTER SWAPPING------------\n "); printf("a=");disp(p); printf("\nb=");disp(q);

return 0;

}

//7. Write a C program to reverse an array using pointers.

#include<stdio.h> void disp(int \*a)

{ for(int i=0;i<5;i++) printf("%d ",\*(a+i));

} int main() { int a[10],i,n=5,j,t; int \*p; p=a; printf("Enter the 5 elements of array: "); for(i=0;i<5;i++) scanf("%d",p+i);

printf("\n----------ACTUAL ARRAY------------\n "); disp(p);

for(i=0,j=n-1;i<=n/2;i++)

{ t=\*(p+i); \*(p+i)=\*(p+j); \*(p+j)=t; j--; } printf("\n----------REVERSE ARRAY------------\n "); disp(p);

return 0;

}

//8. Write a C program to search an element in array using pointers.

#include<stdio.h> int main() { int a[10],i,n=5,t; int \*p; p=a; printf("Enter the 5 elements of array: "); for(i=0;i<n;i++) scanf("%d",p+i); printf("\nEnter the element to be search: "); scanf("%d",&t); for(i=0;i<n;i++)

{ if(\*(p+i)==t)

{ printf("\nElement is present at %d index!",i); break;

} } if(i==n) { printf("\nElement not found");

} return 0;

}

// . What will be the output of the C program?

#include<stdio.h> void function(char\*\*); int main() {

char \*arr[] = { "ant", "bat", "cat", "dog", "egg", "fly" }; function(arr); return 0; }

void function(char \*\*ptr)

{ char \*ptr1;

ptr1 = (ptr += sizeof(int))[-2]; printf("%s\n", ptr1);

}

OUTPUT: cat

If we are using 32bit architecture the int will be of 4 byte so firstly ptr will start pointing to 4th index’s address(egg’s) then ptr1 will be assign with -2 index that is 2nd index whose value is cat

Practical 10: Program to implement or use double pointers and array of pointers concepts in C language.

//1. Write C program to demonstrate pointer to pointer or double pointer.

#include<stdio.h> void main ()

{ int a = 10; int \*p; int \*\*dp; p = &a; // pointer p is pointing to the address of a dp = &p; // pointer dp is a double pointer pointing to the address of pointer p printf("address of a: %x\n",p); // Address of a will be printed printf("address of p: %x\n",dp); // Address of p will be printed printf("value stored at p: %d\n",\*p); // value stoted at the address contained by p i.e. 10 will be printed printf("value stored at pp: %d\n",\*\*dp); // value stored at the address contained by the pointer stoyred at pp

}

//2. Write C program to demonstrate array using pointer to pointer or double pointer.

#include<stdio.h> void main ()

{ int \*p; int \*\*dp; int a[5],i;

p = a; dp = &p; printf("Enter 5 elements for array: "); for(i=0;i<5;i++)

{

scanf("%d",\*dp+i);

} printf("\nYour array is as follows:s ");

printf("%d ",\*\*dp+i);

}

//3. Write a program in C to perform addition of matrix on 2D array.

#include<stdio.h> void add\_matrix(int \*a,int \*b,int \*x, int r,int c)

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

{ for(j=0;j<c;j++)

\*((x+i\*m)+j)=(\*((a+i\*m)+j))+(\*((b+i\*m)+j)); m+=10;

} } void display(int \*a, int rows, int cols)

{

int i, j,m=0;

for (i = 0; i < rows; i++)

{ for (j = 0; j < cols; j++)

{ printf("%d ", \*((a+i\*m)+j));

} m+=10;

printf("\n");

} } void input(int \*a,int r,int c)

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

{

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

{ scanf("%d",((a+i\*m)+j));

} m=m+10;

} } int main() {

int a[10][10],b[10][10],x[10][10], i,j,r,c;

printf("Enter the number of rows: "); scanf("%d",&r); printf("\nEnter the number of coloumns: "); scanf("%d",&c); printf("\nEnter %d values for 1st matrix: ",r\*c); input(a,r,c);

/\*since both the matrix should me equivalent for addition therefor I'm using the same row and column for 2nd matrix as well\*/ printf("\nEnter %d values for 2st matrix: ",r\*c); input(b,r,c); add\_matrix(a,b,x,r,c); //calling additoin function for x=a+b printf("\n-------1st matrix------------\n"); display(a,r,c);

printf("\n-------2nd matrix-------------\n"); display(b,r,c);

printf("\n-------resultant matrix------------\n"); display(x,r,c);

return 0;

}

Practical 11: Program to implement string manipulation functions in C language

//1. Write C program to demonstrate string using pointer to pointer or double pointer.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main() { char \*\*words;

words = (char \*\*) malloc(5 \* sizeof( \* words));

words[0] = "apple"; words[1] = "banana"; words[2] = "jackfruit"; words[3] = "melon"; words[4] = "orange";

for (int i = 0; i < 5; i++) { printf("%s\n", words[i]);

} free(words); return 0;

}

//2. Write a program to find length of string using string array.

#include <stdio.h>

#include<stdlib.h>

#include <stdlib.h> #include <string.h> int get\_length(char \*s)

{ int i; for(i=0;s[i]!='\0';i++); return i;

} int main() { char \*\*str; int i; str=(char\*\*)malloc(5\*sizeof(\*str)); printf("Enter five names: "); for(i=0;i<5;i++) gets(str+i); printf("\nThe lenght of your each word is:\n"); for (i = 0; i < 5; i++) { printf("%d word length is: %d\n",i+1,get\_length(str+i));

}

return 0;

}

//3. Write a program to copy one string to another using string array.

#include <stdio.h>

#include<stdlib.h> #include <string.h> void copy(char \*s,char \*t)

{ int i,l=get\_length(s); for(i=0;i<l;i++)

\*(t+i)= \*(s+i);

\*(t+i)='\0';

}

int get\_length(char \*s)

{ int i; for(i=0;s[i]!='\0';i++); return i;

} int main() { char str[5][20],temp[20]; int i,n; printf("Enter five names:\n "); for(i=0;i<5;i++)

{ scanf("%s",(str+i)); fflush(stdin);

}

printf("\nEnter the word you want to add in the list: "); gets(temp); printf("Enter the position at which you want to add: "); scanf("%d",&n); copy(temp,str+n-1); printf("\nUpdated list: "); for(i=0;i<5;i++) printf("%s\n",(str+i));

return 0;

}

//4. Write a program to concatenate two strings using string array.

#include <stdio.h>

#include<stdlib.h> #include <string.h>

void concatenate(char s[],char t[])

{ int j=0,i,l=get\_length(t); for(i=l;s[j]!='\0';j++)

{

\*(t+i)= \*(s+j); i++;

}

\*(t+i)='\0';

}

int get\_length(char \*s)

{ int i; for(i=0;s[i]!='\0';i++); return i;

} int main() { char str[5][20],temp[20]; int i,n; printf("Enter five names:\n "); for(i=0;i<5;i++)

{ scanf("%s",(str+i)); fflush(stdin);

}

printf("\nEnter the word you want to add in the list: "); gets(temp); printf("Enter the position at which you want to add: "); scanf("%d",&n); concatenate(temp,str+n-1); printf("\nUpdated list: "); for(i=0;i<5;i++) printf("%s\n",(str+i));

return 0;

}

//5. Write a program to compare two strings using string array.

#include <stdio.h>

#include<stdlib.h> #include <string.h>

int compare(char s[],char t[5][20])

{ int j,i,sl=get\_length(s),tl; for(i=0;i<5;i++)

{ tl=get\_length(t+i); if(sl!=tl) continue;

for(j=0;s[j]!='\0';j++)

{

if(s[j]!=t[i][j])

{ break;

} } if(j==sl) { printf("Word is presnt at %d number",i+1); break;

} } return 1; } int get\_length(char \*s)

{ int i; for(i=0;s[i]!='\0';i++); return i;

} int main() { char str[5][20],temp[20]; int i,n;

printf("Enter five names:\n "); for(i=0;i<5;i++)

{

scanf("%s",(str+i)); fflush(stdin);

} printf("\nEnter the word you want to search in the list: "); gets(temp); n=compare(temp,str); if(n) { } else { printf("\nNot found"); } printf("\nUpdated list: "); for(i=0;i<5;i++) printf("%s\n",(str+i));

return 0; }

Practical 12: Program to implement structure and union in C language

//1. Write C program to create, declare and initialize structure.

#include<stdio.h> struct example{ //Creating a structure as example int a; char b; }; void main() { struct example obj; //Declareing it's object obj.a=10; //Initializing it's object obj.b='m'; printf("a=%d\nb=%c",obj.a,obj.b);

}

//2. Write C program to read and print an employee's detail using structure.

#include<stdio.h> struct employee{ //Creating a structure as example int id; char name[20]; float age; }; void main() { struct employee obj; //Declareing it's object printf("Enter employee id: "); scanf("%d",&obj.id); printf("Enter employee name: "); scanf("%s",&obj.name); printf("Enter employee age: "); scanf("%f",&obj.age); printf("\n--------Employee's Details are as follows-------"); printf("\nName:%s\nAGe:%f\nId:%d",obj.name,obj.age,obj.id);

}

//3. Write C program to demonstrate example of nested structure.

#include<stdio.h> struct employee{ //Creating a structure as example int id; struct name{ char first[20]; char last[20]; }ename; float age; }; void main() { struct employee obj; //Declareing it's object printf("Enter employee id: "); scanf("%d",&obj.id); printf("Enter employee first name: "); scanf("%s",obj.ename.first);

printf("Enter employee last name: "); scanf("%s",obj.ename.last); printf("Enter employee age: "); scanf("%f",&obj.age); printf("\n--------Employee's Details are as follows-------"); printf("\nName:%s

%s\nAge:%.1f\nId:%d",obj.ename.first,obj.ename.last,obj.age,obj.id);

}

//4. Write C program to demonstrate example structure pointer

#include<stdio.h> struct student{ int roll; char name[40]; float marks;

}; void main() { struct student obj,\*ptr; ptr=&obj; printf("Enter student roll: "); scanf("%d",&(ptr->roll)); printf("Enter student's name: "); fflush(stdin); gets(ptr->name);

printf("Enter student marks: "); scanf("%f",&(ptr->marks)); printf("\n--------Student's Details are as follows-------");

printf("\nRoll No.:%d\nName:%s\nmarks:%.1f",(ptr->roll),(ptr->name),(ptr-

>marks));

}

//5. Write C program to declare, initialize a union, example of union.

#include<stdio.h> #include<string.h> union example{ int roll; char name[40]; float marks;

}; void main() { union example obj; obj.roll=342;

printf("Roll number : %d",obj.roll); strcpy(obj.name,"rahul"); printf("\nName is: %s",obj.name); obj.marks=450.00; printf("\nMarks is : %.2f",obj.marks);

}

//6. Write C program to demonstrate example of structure of array.

#include<stdio.h> #include<string.h> struct student{ int roll; char name[40]; int marks[5];

}; void main() { struct student s[30]; int i,t[30]={0},j;

printf("Enter the detai;s for 30 students: \n"); for(i=0;i<2;i++)

{ printf("\nEnter student %d details\n",i+1); printf("Enter roll number: "); scanf("%d",&s[i].roll); printf("Enter name: "); fflush(stdin); gets(s[i].name); printf("\nEnter marks in 5 subjects:\n "); for(j=0;j<5;j++)

{ printf("Enter sub %d marks:",j+1); scanf("%d",&s[i].marks[j]); t[i]+=s[i].marks[j];

} } printf("\nStudents details are as follows:"); printf("\nName\t\tRoll\ttotal-marks\n"); for(i=0;i<2;i++)

{

printf("\n%s\t%d\t%d",s[i].name,s[i].roll,t[i]);

}

}

// 7. Write C program to show different between structure and union.

#include <stdio.h>

#include <string.h>

// declaring structure struct struct\_employee

{ int id; float age; char name[20];

};

// declaring union

union union\_employee

{ int id; float age; char name[20];

};

void main()

{ struct struct\_employee s={18,38,"LearnC"};

union union\_employee u={18,38,"LearnC"};

/\*difference one: here we will see that all the data of structure object will be display but in case of union the value of employee id will be display only rest will contain garbage value\*/ printf("structure data:\n id: %d\n" "age: %.2f\n name: %s\n",

s.id, s.age, s.name); printf("\nunion data:\n id: %d\n" "age: %.2f\n name: %s\n",

u.id, u.age, u.name);

/\* difference two : size of structure's object is sum of all its members size where as in case of union the size it's object will the size of it's largest member\*/

printf("\nsizeof structure : %d\n", sizeof(s)); printf("sizeof union : %d\n", sizeof(u));

/\* difference three: we can access all the members of structure at once but can't do the same in case of union\*/ printf("\n Accessing all members at a time:"); s.id = 183;

s.age = 32.5;

strcpy(s.name, "LearnC");

printf("structure data:\n id: %d\n " "age: %.2f\n name: %s\n",

s.id, s.age, s.name);

u.id = 183;

u.age = 32.5;

strcpy(u.name, "LearnC");

printf("\nunion data:\n id: %d\n " "age: %.2f\n name: %s\n",

u.id, u.age, u.name);

//for union we can access one member at a time printf("\n Accessing one member at time:");

printf("\n union data:"); u.id = 240; printf("\ninteger: %d", u.id);

u.age = 12.0; printf("\ndecimal: %f", u.age);

strcpy(u.name, "XYZ"); printf("\nname: %s\n", u.name);

}

//8.Write C program to use typedef for creating structure and union.

#include <stdio.h>

#include <string.h>

typedef struct student

{ int id; char name[20]; float percentage;

} status; //here we have given an allias status to "struct student"

void main() { status record; record.id=123; strcpy(record.name, "Mohan kumar"); record.percentage = 76.5; printf("Student record is:\n"); printf(" Id : %d \n", record.id); printf(" Name : %s \n", record.name);

printf(" Percentage : %f \n", record.percentage);

}

Practical 13: Program to implement macro and storage classes in C language

/\*1. Write C program to define the math operator ‘+’ as PLUS, ‘-‘ as MINUS,

‘\*’ as MULT & ‘/’ as DIVIDE using pre-processor directives and do the operations over variables

(x, y) defined on above question like z=x PLUS y.\*/

#include<stdio.h>

#define PLUS +

#define MINUS -

#define MULT \* #define DIVIDE / void main()

{ int a=6,b=2; printf("a=%d and b=%d\n",a,b); printf("a+b=%d\n",a PLUS b); printf("a-b=%d\n",a MINUS b); printf("a\*b=%d\n",a MULT b); printf("a/b=%d",a DIVIDE b);

}

/\*2. Write C program to define a macro with one parameter to compute the volume of a sphere.

Write a C program using this macro to compute the volume for spheres of radius

5, 10 and 15 meters.\*/

#include<stdio.h>

#define volume(r) ((4.0/3) \* 3.1415\*r\*r\*r) void main()

{

int radius=5;

printf("volume of sphere of radius 5 is %f\n",volume(radius)); radius=10;

printf("volume of sphere of radius 10 is %f\n",volume(radius)); radius=15;

printf("volume of sphere of radius 15 is %f\n",volume(radius));

}

/\*3. Write C program to define a macro that receives an array and the number of elements in the array as arguments. Write a C program for using this macro to print the elements of the array.\*/

#include<stdio.h>

#define display(array, length) \ for(int i = 0; i < length; i++) \ printf("%d\t", array[i]);

int main(void) { int array[5] = {4, 2, 3, 1, 0}; printf("Your array is as follows: "); display(array, 5); return 0;

}

/\*4. Write C program to illustrate the properties of a static variable.\*/

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

int Myfunc();

// value of the static variable persists within the function call

//so each time we will get the updated value printf("%d",Myfunc()); printf("\n%d",Myfunc()); printf("\n%d",Myfunc());

return 0;

} int Myfunc() { static int count=0; count++; return count;

}

//5. Write C program to illustrate the properties of an auto variable.

#include <stdio.h> int main() { auto int a = 10,i; printf("%d ",++a);

{ auto int a = 20; printf("%d ",a); // 20 will be printed since it is the local value of a } printf("%d ",a); // 11 will be printed since the scope of a = 20 is ended. }

/\*6. Write C program to illustrate the properties of an extern variable.\*/

#include <stdio.h>

int main() { extern int a; // Compiler will search here for a variable a defined and initialized somewhere in the pogram or not.

printf("%d",a);

}

int a = 20;

//7. Write C program to illustrate the properties of a register variable.

#include <stdio.h> int main() { register int i; //here I have declared i as register variable because I wanna use it as iteration variable for(i=0;i<10;i++)

{ printf("%d ",i);

}

}

Practical 14: Program to perform operations using malloc, calloc, realloc, free and on file handling in C language.

//1. Write C program to read file using fgetc(), fgets(), and fscanf() functions. #include <stdio.h> #include<stdlib.h> int main() {

FILE \*ptr; char ch,str[100]; ptr=fopen("myfile.c","r"); if(ptr==NULL)

{

printf("unable to open file!"); return 0;

} printf("Reading file with the help of fgetc():\n"); while((ch=fgetc(ptr))!=EOF)//fgetc() is used to obtain input from a file single character at a time.

{ printf("%c",ch);

} fclose(ptr); ptr=fopen("myfile.c","r");

printf("\nReading file with the help of fgets():\n"); while(fgets(str,30,ptr)!=NULL)//fgets() is used to obtain input from a file a number of character (as specified) at a time.

{

printf("%s",str);

} fclose(ptr); ptr=fopen("myfile.c","r");

printf("\nReading file with the help of fsacnf():\n"); while(fscanf(ptr,"%s",str)!=EOF)/\*fscanf() reads a word from the file and returns EOF at the end of file.\*/

{

printf("%s ",str);

} return 0;

}

//2. Write C program to write data into any file using fputc(), fputs(), and fprintf() functions. #include <stdio.h> #include<stdlib.h> int main() {

FILE \*ptr; char ch,str[100]; ptr=fopen("myfile.c","w"); if(ptr==NULL)

{ printf("unable to open file!"); return 0;

} printf("Writing file with the help of fputc():\n"); while((ch=getchar())!='$')

{ fputc(ch,ptr);

} fclose(ptr);

ptr=fopen("myfile.c","a");

printf("Writing file with the help of fputs():\n"); while(gets(str)!=NULL)

{ fputs(str,ptr);

} printf("\nWriting file with the help of fprintf():\n"); while(gets(str)!=NULL)

{ fprintf(ptr, "%s",str);

}

printf("\nfile created successfully!"); return 0;

}

//3. Write a program to copy one file to another.

#include <stdio.h>

#include <stdlib.h>

int main()

{

FILE \*source, \*dest; char filename[100], ch;

printf("Enter the filename to open for reading \n"); scanf("%s", filename);

source = fopen(filename, "r"); if (source == NULL)

{ printf("Cannot open file %s \n", filename); exit(0);

} printf("Enter the filename to open for writing \n"); scanf("%s", filename);

// Open another file for writing dest = fopen(filename, "w"); if (dest == NULL)

{ printf("Cannot open file %s \n", filename); exit(0);

}

// Read contents from file

while (ch = fgetc(source) != EOF)

{

fputc(ch, dest);

}

printf("\nContents copied to %s", filename);

fclose(source); fclose(dest); return 0;

}

//4. Write C program to create dynamic array using malloc() function.

#include <stdio.h>

#include <stdlib.h>

int main() { int \*ptr,n,i,total=0; printf("Enter the number of element: "); scanf("%d",&n); ptr=(int\*)malloc(n\*sizeof(n)); if(ptr==NULL)

{

printf("\nUnable allocate memory"); return 0;

} printf("\nEnter %d elemets: ",n); for(i=0;i<n;i++)

{ scanf("%d",ptr+i); total+=\*(ptr+i);

} free(ptr); printf("\nAverage is: %.2f",(float)total/n); return 0;

}

//5. Write C program to create dynamic array using calloc() function.

#include <stdio.h>

#include <stdlib.h>

int main() { int \*ptr,n,i,total=0; printf("Enter the number of element: "); scanf("%d",&n); ptr=(int\*)calloc(n,sizeof(n)); if(ptr==NULL)

{

printf("\nUnable allocate memory"); return 0;

} printf("\nEnter %d elemets: ",n); for(i=0;i<n;i++)

{ scanf("%d",ptr+i); total+=\*(ptr+i);

} free(ptr); printf("\nTotal is: %d",total); return 0;

}

//6. Write C program to create dynamic array and re-size array using realloc() function. #include <stdio.h>

#include <stdlib.h>

int main() { int \*ptr,n,i,total=0,n2; char ch; printf("Enter the number of element: "); scanf("%d",&n); ptr=(int\*)malloc(n\*sizeof(n)); if(ptr==NULL)

{

printf("\nUnable allocate memory"); return 0;

} printf("\nEnter %d elemets: ",n); for(i=0;i<n;i++)

{ scanf("%d",ptr+i); total+=\*(ptr+i);

} printf("\nDo you want to add more?(y/n) "); fflush(stdin); scanf("%c",&ch); if(ch=='y')

{ printf("\nHow many element: "); scanf("%d",&n2); ptr=(int\*)realloc(ptr,n+n2); printf("\nEnter %d element: ",n2); for(;i<n+n2;i++)

{ scanf("%d",ptr+i); total+=\*(ptr+i);

} } free(ptr); printf("\nTotal is: %d",total); return 0;

}

Practical 15: Program to perform graphical operations in C language.

//1. Write C program to draw line.

#include <graphics.h> #include <conio.h> void main() { int gd = DETECT, gm;

initgraph(&gd, &gm, "C:\\TC\\BGI");

line(100, 100, 200, 200);// draws a line from (100, 100) to (200, 200)

getch(); closegraph();

}

//2. Write C program to draw circle

#include<graphics.h>

#include<stdio.h> #include<conio.h> void main() { int gd=DETECT,gm; int x,y,radius; clrscr(); initgraph(&gd,&gm,"C:\\TURBOC3\\BGI"); printf("\n Please enter the centre coordinate of the circle: "); scanf("%d %d",&x,&y); printf("\nPlease enter the radius of the circle: "); scanf("%d",&radius); printf("\n\*\*\*\*\* CIRCLE \*\*\*\*\*\*\n"); circle(x,y,radius); getch(); closegraph();

}

//3. Write C program to draw rectangle.

#include<graphics.h>

#include<stdio.h> #include<conio.h> void main() { int gd=DETECT,gm; int top,left,bottom,right; clrscr(); initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

left = 100; top = 80; bottom = 220; right = 350;

printf("\n\*\*\*\*\* Rectangle \*\*\*\*\*\*\n"); rectangle(left, top, right, bottom); getch(); closegraph();

}

//4.Write C program to move circle one location to another on pressing enter key.

#include<graphics.h>

#include<stdio.h> #include<conio.h> void main() { int gd=DETECT,gm,i=1; clrscr();

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI"); while(ch=='\n')

{

circle(200+(++i\*10),200,50); delay(50); cleardevice();

}

closegraph();

}