### Practical File

### Of

### Fundamentals Of C Programming

### 23CS003

#### Submitted

#### in partial fulfillment for the award of the degree of

## BACHELEOR OF ENGINEERING

***in***

****COMPUTER SCIENCE & ENGINEERING

**CHITKARA UNIVERSITY**

**CHANDIGARH-PATIALA NATIONAL HIGHWAY**

**RAJPURA (PATIALA) PUNJAB-140401 (INDIA)**

##### January, 2024

##### **Submitted To: Submitted by:**

##### Faculty Name:- Dr.Monika Sethi Name:-Swayam Kataria

##### Designation:- Assiociate Professor Roll no.:2310990980

Chitkara University, Punjab Sem, Batch:- 2nd, 2024

1#wap to depict printf/scanf:

#include <stdio.h>

int main()

{ int a,b;

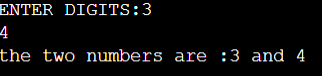
printf("ENTER DIGITS:");

scanf("%d%d",&a,&b);

printf("the two numbers are :%d and %d",a,b);

return 0;

}



2#WAP to add two no and display their sum:

#include<stdio.h>

int main()

{

int a, b, sum;

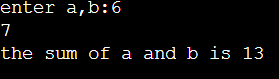
printf("enter a,b:");

scanf("%d%d",&a,&b);

printf("the sum of a and b is %d",sum=a+b,a,b);

return 0;

}



3#WAP to find area and circumference of circle:

#include<stdio.h>

int main()

{

int radius;

float area,circumference;

float pi=3.14;

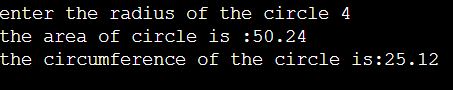
printf("enter the radius of the circle ");

scanf("%d",&radius);

printf("the area of circle is :%.2f\n",area=pi\*radius\*radius,pi,radius);

printf("the circumference of the circle is:%.2f",circumference=2\*pi\*radius,pi,radius);

}



4#WAP to perform add,sub,div,mul:

#include <stdio.h>

int main()

{

int a,b,sum,mul,div,sub;

printf("enter a and b");

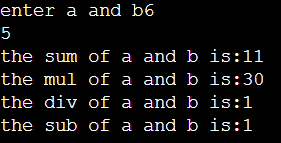
scanf("%d%d",&a,&b);

printf("the sum of a and b is:%d\n",sum=a+b,a,b,sum);

printf("the mul of a and b is:%d\n",mul=a\*b,a,b,mul);

printf("the div of a and b is:%d\n",div=a/b,div,a,b);

printf("the sub of a and b is:%d\n",sub=a-b,a,b,sub);

} 

5#WAP TO EVALUATE :

i)v=u+at:

#include<stdio.h>

int main()

{

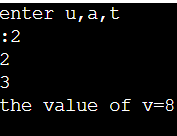
int v,u,a,t;

printf("enter u,a,t\n:");

scanf("%d%d%d",&u,&a,&t);

printf("the value of v=%d",v=u+a\*t,u,a,t);

}



ii)s=ut+1/2at\*t

#include<stdio.h>

int main()

{ float s;

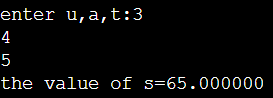
int u,a,t;

printf("enter u,a,t:");

scanf("%d%d%d",&u,&a,&t);

printf("the value of s=%2f",s=u\*t+0.5\*a\*t\*t,s,u,a,t);

}



iii)T=2\*a+b^1/2+9c:

#include<stdio.h>

#include<math.h>

int main()

{

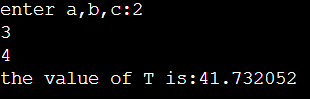
float T,a,c,b;

printf("enter a,b,c:");

scanf("%f%f%f",&a,&b,&c);

printf("the value of T is:%f",T=2\*a+sqrt(b)+9\*c,a,b,c);

}



iv)h=sqrt(p^2+b^2):

#include<stdio.h>

#include<math.h>

int main()

{

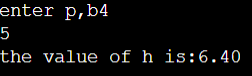
float h,p,b;

printf("enter p,b");

scanf("%f%f",&p,&b);

printf("the value of h is:%.2f",h=sqrt(b\*b+p\*p));

}



6#WAP to swap 2 variable

1. Temp:

#include<stdio.h>

int main()

{

int a,b,temp;

printf("enter a,b:");

scanf("%d%d",&a,&b);

temp=a;

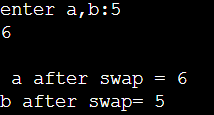
a=b;

b=temp;

printf("\n a after swap = %d\n", a);

printf("b after swap= %d", b);

}



b)without swap

#include<stdio.h>

int main()

{

int a, b;

printf("enter a,b");

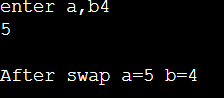
scanf("%d%d",&a,&b);

a=a+b;

b=a-b;

a=a-b;

printf("\nAfter swap a=%d b=%d",a,b);

return 0; }

7#WAP to find greatest among three:

a)Conditional op:

#include<stdio.h>

int main()

{

int a,b,c,greatest;

printf("\nEnter 3 numbers:");

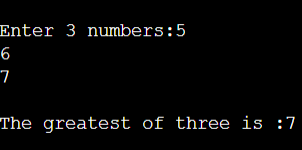
scanf("%d %d %d",&a,&b,&c);

greatest=(a>b&&a>c?a:b>c?b:c);

printf("\nThe greatest of three is :%d",greatest);

return 0;

}



b)If else statement:

#include<stdio.h>

int main()

{

int a,b,c;

printf("enter three numbers:");

scanf("%d%d%d",&a,&b,&c);

if (a >= b && a>= b)

{

printf("%d is the largest number.", a);

}

else if (b >= a && b >= c)

{

printf("%d is the largest number.", b);

}

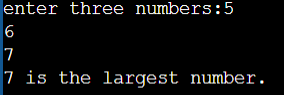
else if (c >= a && c >= b)

{

printf("%d is the largest number.", c);

}

}



8# WAP using switch statement:

1. To check that input number is vowel or consonant

#include <stdio.h>

int main()

{

char ch;

printf("Enter an alphabet: ");

scanf("%c", &ch);

switch(ch)

{

case 'a':

printf("Vowel");

break;

case 'e':

printf("Vowel");

break;

case 'i':

printf("Vowel");

break;

case 'o':

printf("Vowel");

break;

case 'u':

printf("Vowel");

break;

case 'A':

printf("Vowel");

break;

case 'E':

printf("Vowel");

break;

case 'I':

printf("Vowel");

break;

case 'O':

printf("Vowel");

break;

case 'U':

printf("Vowel");

break;

default:

printf("Consonant");

}

return 0;

} 

b)to check whether a number is positive , negative or zero:

#include <stdio.h>

int main()

{

int num;

printf("Enter any number: ");

scanf("%d", &num);

switch (num > 0)

{

// Num is positive

case 1:

printf("%d is positive.", num);

break;

case 0:

switch (num < 0)

{

case 1:

printf("%d is negative.", num);

break;

case 0:

printf("%d is zero.", num);

break;

}

break;

}

return 0;

}



9:WAP to print sum of n natural numbers using while loop:

#include<stdio.h>

int main()

{

int i=0,n,sum=0;

printf("enter n:\n");

scanf("%d",&n);

while(i<=n)

{

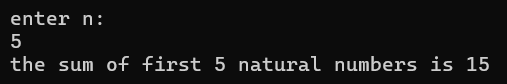
sum=sum+i;

i=i+1;

}printf("the sum of first %d natural numbers is %d",n,sum);

}

Output:



10 WAP to check Armstrong number using for loop.

#include <stdio.h>

int main() {

int num, originalNum, remainder, result = 0;

printf("Enter a three-digit integer: ");

scanf("%d", &num);

originalNum = num;

while (originalNum != 0)

{

remainder = originalNum % 10;

result += remainder \* remainder \* remainder;

originalNum /= 10;

}

if (result == num)

{

printf("%d is an Armstrong number.", num);

}

else

{

printf("%d is not an Armstrong number.", num);

}

return 0;

}

Output



11WAP to print digits of a number and its reverse

#include <stdio.h>

int main() {

int num, count = 0, rev = 0, digit;

printf("Enter a number: ");

scanf("%d", &num);

int temp = num;

while (temp != 0) {

temp /= 10;

count++;

}

printf("Number of digits: %d\n", count);

temp = num;

while (temp != 0)

{

digit = temp % 10;

rev = rev \* 10 + digit;

temp /= 10;

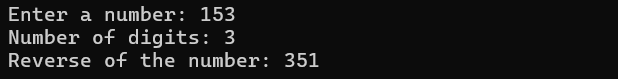
}

printf("Reverse of the number: %d\n", rev);

return 0;

}

Output



12. WAP to print fibonacci series

#include <stdio.h>

int main() {

int num, first\_num = 0, second\_num = 1, next\_num, i;

printf("Enter the number of terms: ");

scanf("%d", &num);

printf("Fibonacci Series:\n");

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

{

printf("%d ", first\_num);

next\_num = first\_num + second\_num;

first\_num = second\_num;

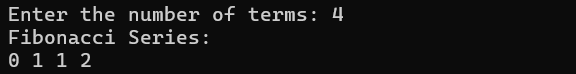
second\_num = next\_num;

}

return 0;

}

Output:



13. WAP to print following patterns:

a) pyramid star:

#include <stdio.h>

int main() {

int i, j, rows;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; ++i) {

for (j = 1; j <= i; ++j) {

printf("\* ");

}

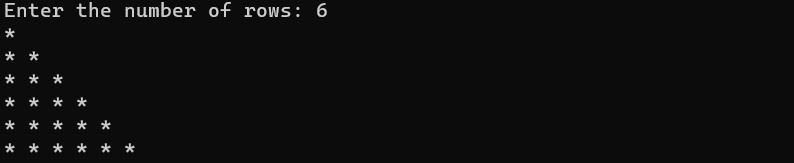
printf("\n");

}

return 0;

}

output



b)inverted pattern pyramid star:

#include <stdio.h>

int main()

{

int i, j, rows;

printf("Enter number of rows: ");

scanf("%d", &rows);

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

{

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

{

printf(" ");

}

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

{

printf("\*");

}

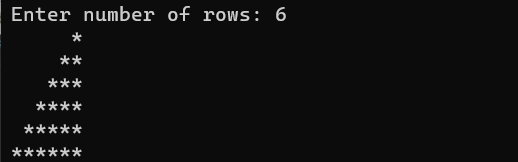
printf("\n");

}

return 0;

}

Output:



14 WAP to print pattern :

#include <stdio.h>

int main()

{

int i, j, N;

printf("Enter number of rows: ");

scanf("%d", &N);

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

{

for(j=1; j<=N; j++)

{

printf("%d ",i\*j);

}

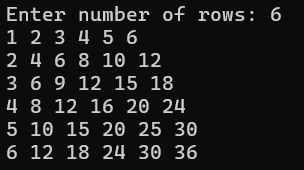
printf("\n");

}

return 0;

}

Output:



13.WAP to check if number is prime , Armstrong or perfect using functions.

#include <math.h>

#include <stdio.h>

int checkPrimeNumber(int n);

int checkArmstrongNumber(int n);

int checkPerfectNumber(int number);

int main() {

int n, flag;

printf("Enter a positive integer: ");

scanf("%d", &n);

flag = checkPrimeNumber(n);

if (flag == 1)

printf("%d is a prime number.\n", n);

else

printf("%d is not a prime number.\n", n);

flag = checkArmstrongNumber(n);

if (flag == 1)

printf("%d is an Armstrong number.", n);

else

printf("%d is not an Armstrong number.", n);

checkPerfectNumber(n);

return 0;

}

int checkPrimeNumber(int n) {

int i, flag = 1, squareRoot;

squareRoot = sqrt(n);

for (i = 2; i <= squareRoot; ++i) {

if (n % i == 0) {

flag = 0;

break;

}

}

return flag;

}

int checkArmstrongNumber(int num) {

int originalNum, remainder, n = 0, flag;

double result = 0.0;

for (originalNum = num; originalNum != 0; ++n) {

originalNum /= 10;

}

for (originalNum = num; originalNum != 0; originalNum /= 10) {

remainder = originalNum % 10;

result += pow(remainder, n);

}

if (round(result) == num)

flag = 1;

else

flag = 0;

return flag;

}

int checkPerfectNumber(int number)

{

int rem, sum = 0, i;

for (i = 1; i <= (number - 1); i++)

{

rem = number % i;

if (rem == 0)

{

sum = sum + i;

}

}

if (sum == number)

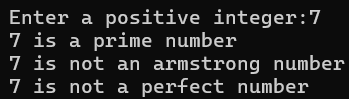
printf("\n%d is perfect number", number);

else

printf("\n%d is not a perfect number", number);

return 0;

}



16.WAP to calculate area and circumference of circle using functions

#include <stdio.h>

float calculateArea(float radius) {

return 3.14 \* radius \* radius;

}

float calculateCircumference(float radius) {

return 2 \* 3.14 \* radius;

}

int main() {

float radius;

printf("Enter the radius of the circle: ");

scanf("%f", &radius);

float area = calculateArea(radius);

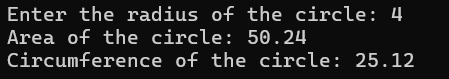
float circumference = calculateCircumference(radius);

printf("Area of the circle: %.2f\n", area);

printf("Circumference of the circle: %.2f\n", circumference);

return 0;

}



17.WAP to swap two variables using the concept of call by value and call by reference.

#include <stdio.h>

void swapByValue(int a, int b) { int temp = a; a = b; b = temp;

}

void swapByReference(int \*a, int \*b) { int temp = \*a; \*a = \*b;

\*b = temp;

}

int main()

{ int num1, num2; swapByValue(num1, num2);

printf("enter two numbers ");

scanf("%d%d",&num1,&num2);

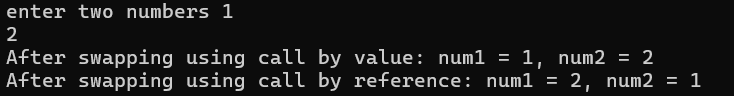
printf("After swapping using call by value: num1 = %d, num2 = %d\n", num1, num2); swapByReference(&num1, &num2);

printf("After swapping using call by reference: num1 = %d, num2 = %d\n", num1, num2);

return 0;

}

Output:



18.WAP to perform the following operations on 1D-Array:

* Insert
* Update
* Delete
* Display
* Search **Solution**:

#include <stdio.h>

#define MAX\_SIZE 100

void insert(int arr[], int \*size, int element, int position) { if (\*size >= MAX\_SIZE || position < 0 || position > \*size) { printf("Invalid insertion.\n"); return;

}

for (int i = \*size; i > position; i--) arr[i] = arr[i - 1];

arr[position] = element;

(\*size)++;

printf("Element inserted successfully.\n");

}

void update(int arr[], int size, int position, int newValue) { if (position < 0 || position >= size) { printf("Invalid position for update.\n"); return;

} arr[position] = newValue;

printf("Element updated successfully.\n");

}

void delete(int arr[], int \*size, int position) { if (position < 0 || position >= \*size) { printf("Invalid position for deletion.\n"); return;

} for (int i = position; i < \*size - 1; i++) arr[i] = arr[i + 1];

(\*size)--;

printf("Element deleted successfully.\n");

}

void display(int arr[], int size) { printf("Array: "); for (int i = 0; i < size; i++) printf("%d ", arr[i]);

printf("\n");

}

int search(int arr[], int size, int element) { for (int i = 0; i < size; i++) if (arr[i] == element) return i;

return -1;

}

int main() {

int array[MAX\_SIZE];

int size = 0, choice, element, position, newValue, searchResult;

do { printf("\n1. Insert\n2. Update\n3. Delete\n4. Display\n5. Search\n6. Exit\nEnter your choice: "); scanf("%d", &choice); switch (choice) { case 1:

printf("Enter element and position to insert: "); scanf("%d %d", &element, &position); insert(array, &size, element, position); break;

case 2:

printf("Enter position and new value to update: "); scanf("%d %d", &position, &newValue); update(array, size, position, newValue); break;

case 3:

printf("Enter position to delete: "); scanf("%d", &position); delete(array, &size, position); break;

case 4: display(array, size); break;

case 5:

printf("enter element to search:");

scanf("%d",&element);

searchResult=search(array,size,element);

if(searchResult1=-1)

printf("element at position:%d\n",searchResult);

else

printf("element dne\n");

break;

case 6:

printf("exit\n");

break;

default:

printf("invalid choice");

}

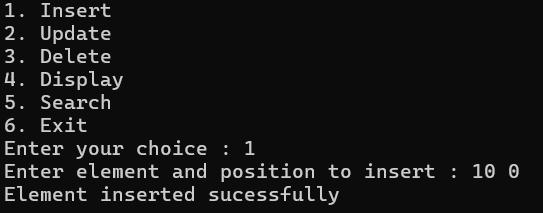
}

while(choice!=6);

return 0;

}

Output:



19. WAP calculate the sum of array elements by passing it to a function.

int sumArray(int arr[], int size) { int sum = 0;

for (int i = 0; i < size; i++) { sum += arr[i];

} return sum;

}

int main()

{ int size;

scanf("%d", &size); int array[size];

printf("Enter %d elements:\n", size); for (int i = 0; i < size; i++) { printf("Element %d: ", i + 1); scanf("%d", &array[i]);

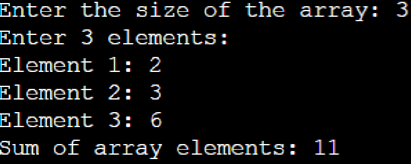
}

int sum = sumArray(array, size);

printf("Sum of array elements: %d\n", sum); return 0;

}

Output



20.WAP to show the use of passing pointer as arguments to the functions.

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

\*b = temp;

}

void increment(int \*num) { (\*num)++;

} int main() { int x, y, number;

printf("Enter two numbers (separated by space): ");

scanf("%d %d", &x, &y);

printf("Before swapping: x = %d, y = %d\n", x, y);

printf("After swapping: x = %d, y = %d\n", x, y);

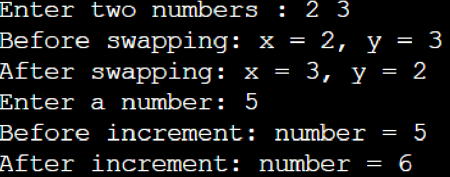
printf("Enter a number: "); scanf("%d", &number);

printf("Before increment: number = %d\n", number);

increment(&number);

printf("After increment: number = %d\n", number); return 0;

}



21.WAP to matrix multiplication using the concept of 2D array.

#include <stdio.h>

#define MAX\_SIZE 10

void matrixMultiplication(int mat1[][MAX\_SIZE], int rows1, int cols1, int mat2[][MAX\_SIZE], int rows2, int cols2, int result[][MAX\_SIZE])

{ for (int i = 0; i < rows1; i++)

for (int j = 0; j < cols2; j++)

{ result[i][j] = 0;

for (int k = 0; k < cols1; k++)

result[i][j] += mat1[i][k] \* mat2[k][j]; }

}

int main()

{ int rows1, cols1, rows2, cols2;

printf("Enter dimensions of the first matrix (rows columns): ");

scanf("%d %d", &rows1, &cols1);

pintf("Enter dimensions of the second matrix (rows columns): ");

scanf("%d %d", &rows2, &cols2); if (cols1 != rows2)

{ printf("Matrix multiplication not possible.\n");

return 1;

}

int mat1[MAX\_SIZE][MAX\_SIZE], mat2[MAX\_SIZE][MAX\_SIZE], result[MAX\_SIZE][MAX\_SIZE];

printf("Enter

elements

of

the

first

matrix:

\

n");

for

(

int

i

=

0

;

i

<

rows1;

i++)

for (int

j = 0; j < cols1; j++)

scanf("%d",

&mat1[i][j]);

printf("Enter

elements

of

the

second

matrix:

\

n");

for

(

int

i

=

0;

i

<

rows2;

i++)

for (int j = 0; j < cols2; j++)

scanf("%d",

&mat2[i][j]);

for

(

int

i

=

0

;

i

<

rows1;

i++)

{

for

(

int

j

=

0

;

j

<

cols2;

j++)

printf("%d

",

result[i][j]);

printf("

\

n");

}

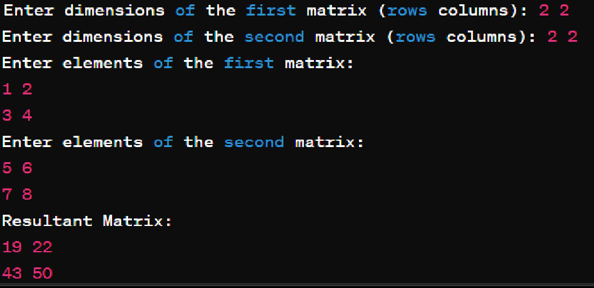
return

0

;

}

**Output:**



22. WAP to transpose a given matrix

#include <stdio.h>

#define MAX\_SIZE 10 int main() { int rows, cols;

printf("Enter dimensions of the matrix (rows columns): ");

scanf("%d %d", &rows, &cols);

int mat[MAX\_SIZE][MAX\_SIZE];

printf("Enter elements of the matrix:\n");

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

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

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

printf("Original Matrix:\n");

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

{ for (int j = 0; j < cols; j++) printf("%d ", mat[i][j]);

printf("\n");

}

printf("\nTransposed Matrix:\n"); for (int j = 0; j < cols; j++) { for (int i = 0; i < rows; i++)

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

printf("\n");

}

Return 0;

}

