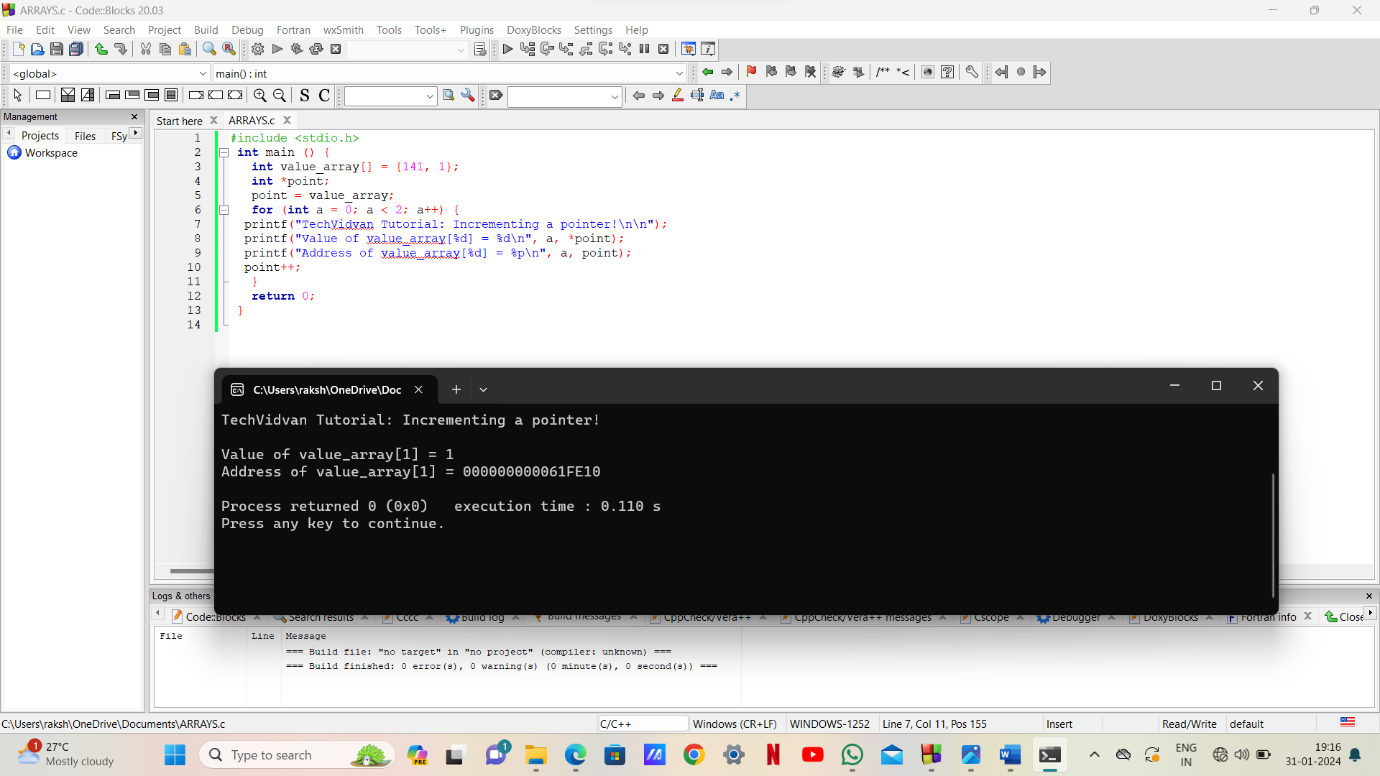
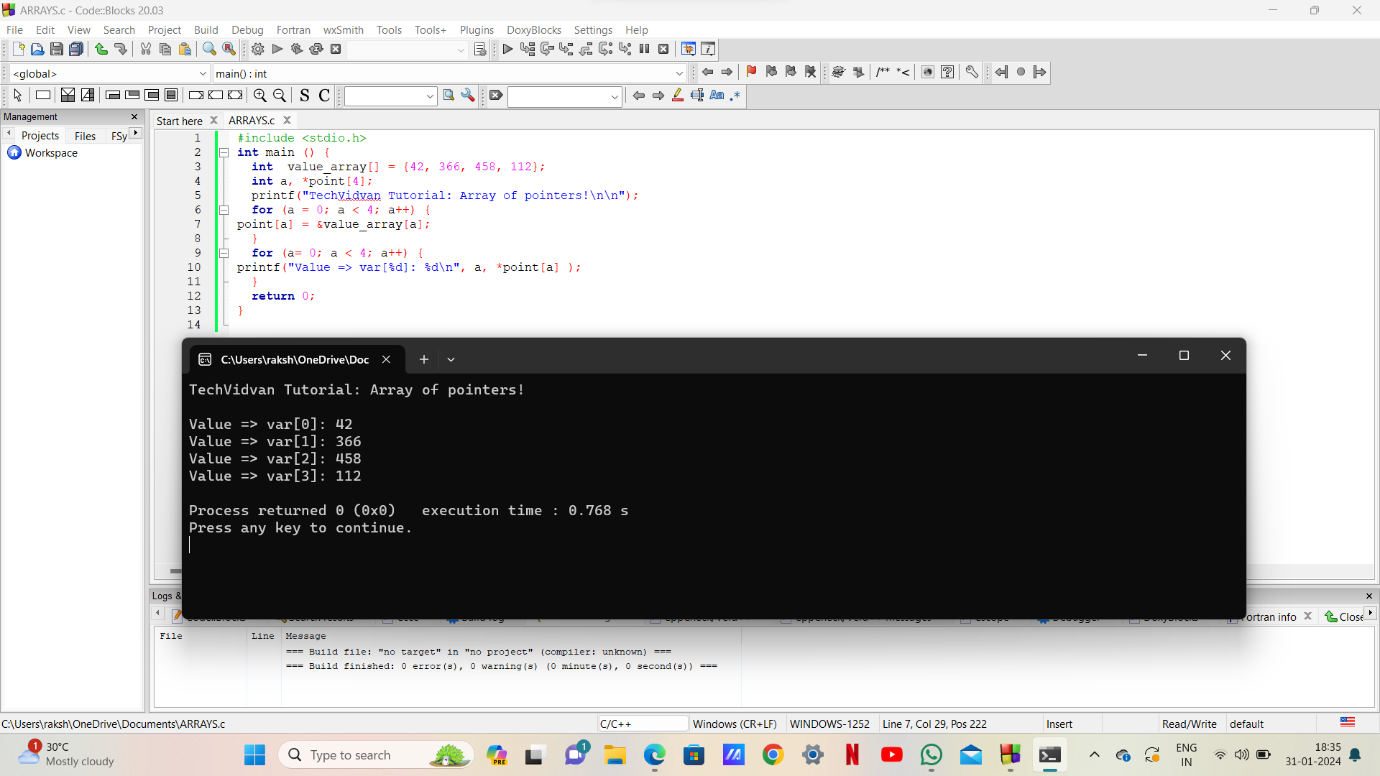
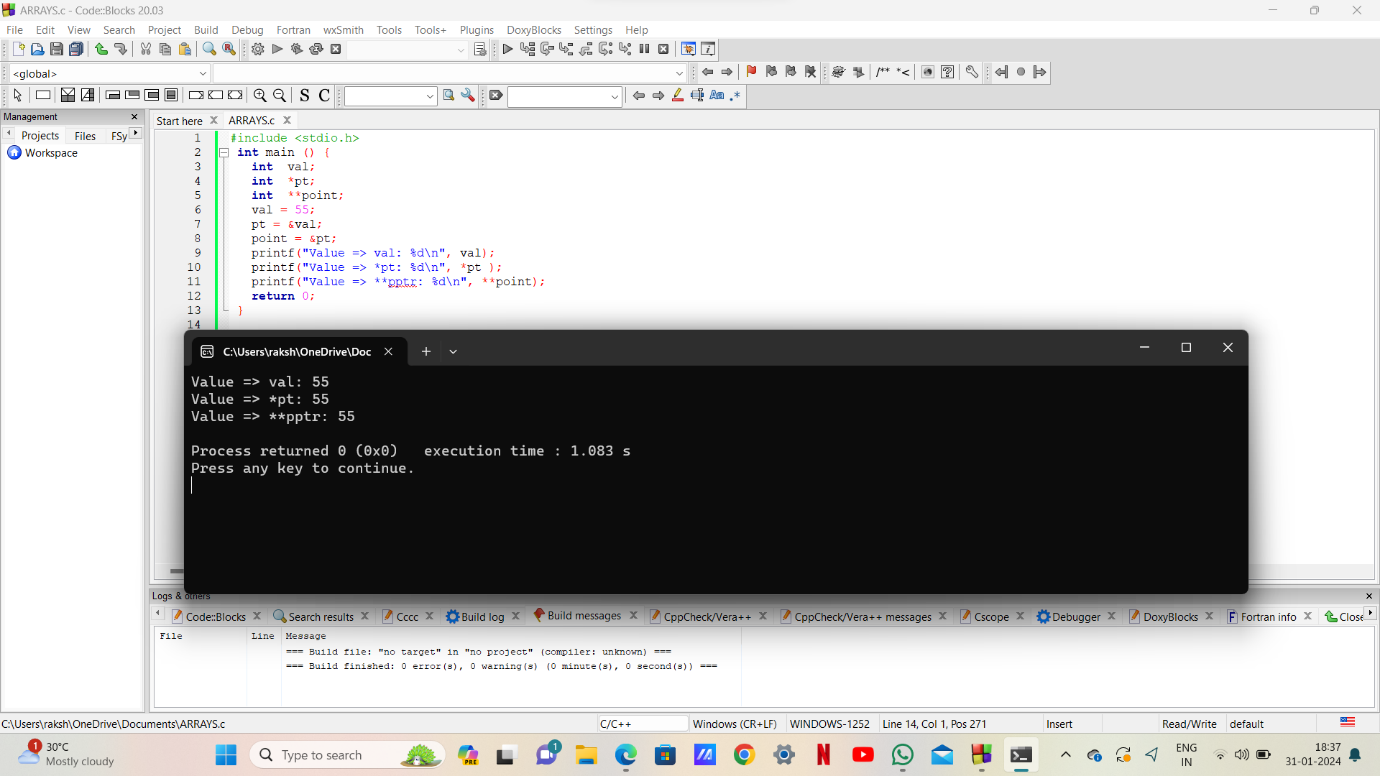
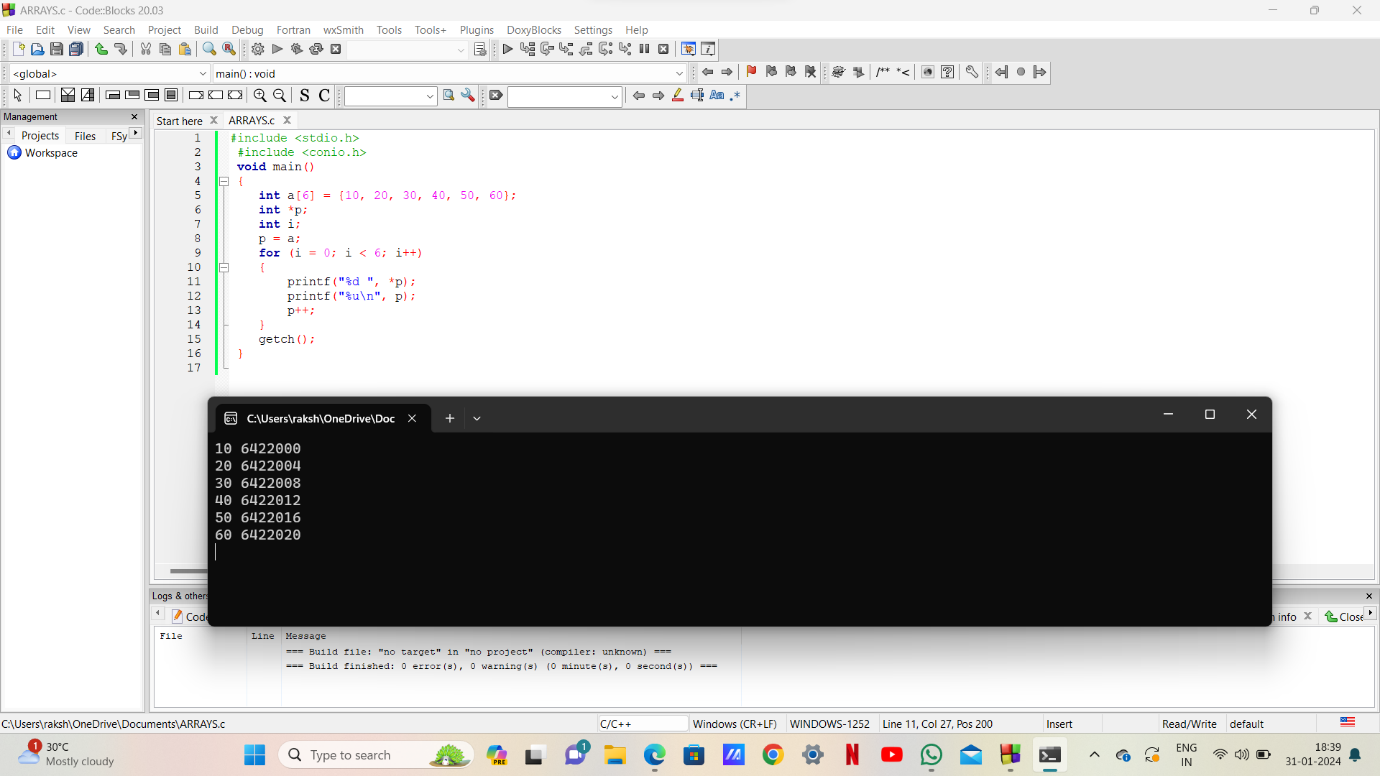
LOOPS AND POINTERS

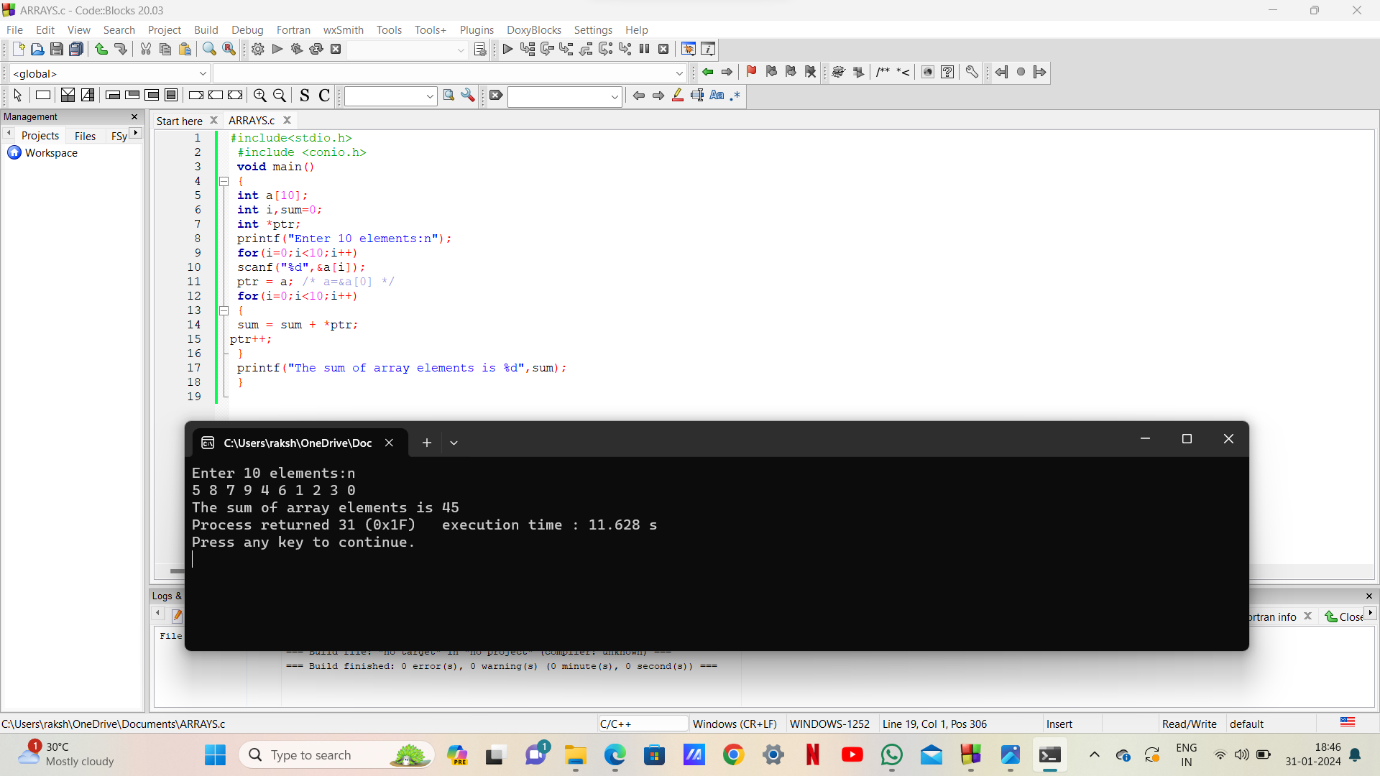
ASSIGNMENT:

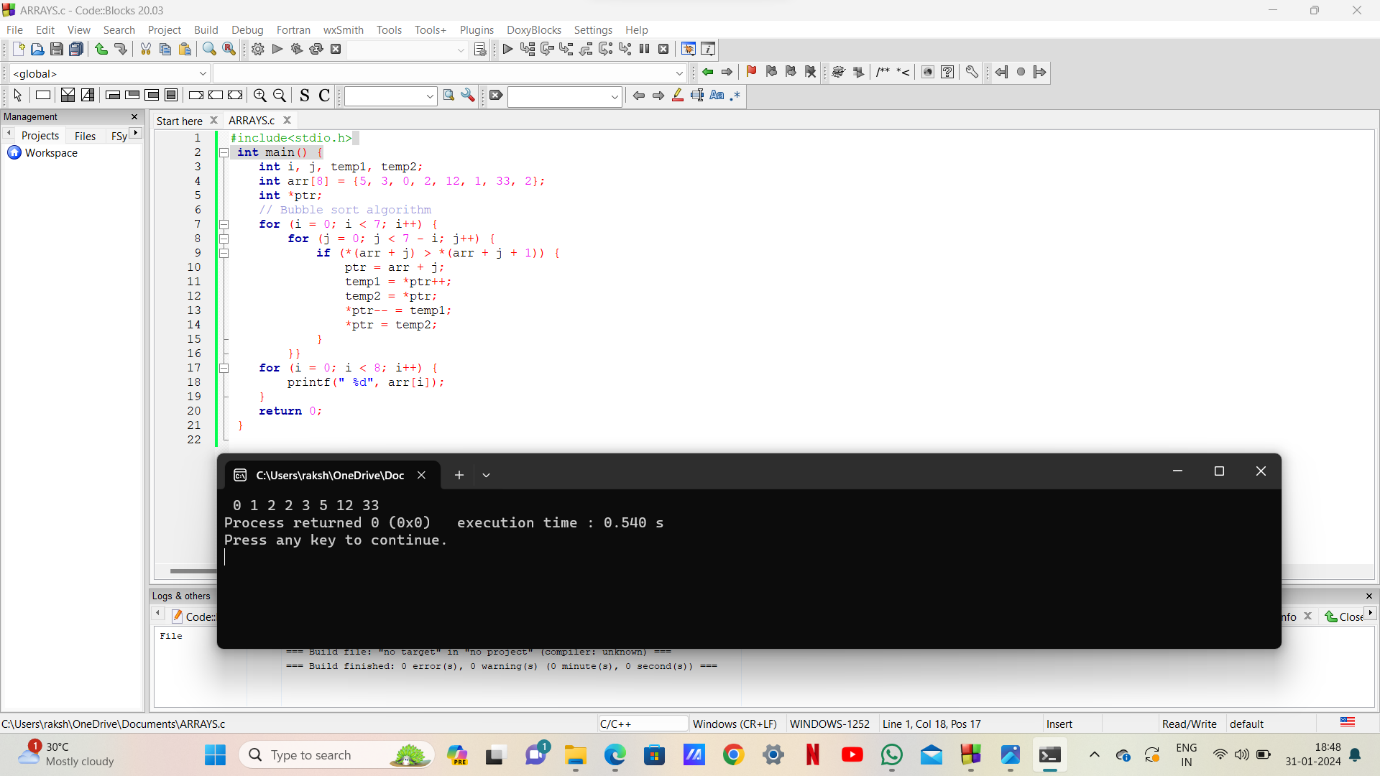












1)

struct node

{

int i;

float j;

};

struct node \*s[10];

The above C declaration defines

GATE CSE 2000

1. An array, each element of which is pointer to a structure of type node

2. A structure of 2 fields, each field being a pointer to an array of 10 elements

3. A structure of 3 fields: an integer, a float, and an array of 10 elements

4. An array, each element of which is a structure of type node

Solution:

The correct option is 1

The above program declares an array of 10 elements where each element is a pointer to the structure of type node the elements in array stores the address of structure

2)

Assume that objects of the type short, float and long occupy 2 bytes, 4 bytes and 8 bytes, respectively. The memory requirement for variable t, ignoring alignment

GATE CSE 2000

Struct

{

short s [5];

union

{

float y;

long z;

}u;

} t;

1. 22 bytes

2. 18 bytes

3. 14 bytes

4. 10 bytes

Solution:

The correct option is 2

The above code declares variable t requies memory of 10 bytes from the short s [5] because short data type occupy of 2 bytes for each.hence for array of 5 elements it occupies 10 bytes and in case of union there are 2 data types float and long occupies 4,8 bytes respectively ,the union takes the size of largest member which is 8 bytes.there fore total memeory reuirement is 10+8=18 bytes.

3)What does the given program print?

char c[ ] = “GATE2011”

char \*p = c;

printf (“%s”, p + p[3] – p[1]);

GATE CSE 2011

1.GATE 2011

2. 2011

3. E2011

4. 01

Solution:

The correct option is 2

The above program shows that p pointer pointing to the above string,p[3] refers character ‘E’ and p[1] refers the character ‘A’.

p[3] – p[1] =69(ASCII value of E) - 65(ASCII value of A) = 4

p+p[3]-p[1] =p+4 -> now the pointer p moves four positions ahead from starting

therefore it prints 2011 .

4)The output of the following C program is\_\_\_\_\_\_\_\_\_\_

GATE CSE 2015 Set 1

void f1(int a, int b) {

int c;

c=a; a=b; b=c;

}

void f2(int \*a, int \*b) {

int c;

c=\*a; \*a=\*b; \*b=c;

}

int main(){

int a=4, b=5, c=6;

f1(a,b);

f2(&b, &c);

printf(“%d”,c-a-b);

}

Output: -5

5)The following program prints \_\_\_\_\_\_\_\_\_\_\_

#include < stdio.h >

void f (int \*p, int \*q) {

p = q;

\*p = 2;

}

int i = 0, j = 1;

int main ( ){

f(&i, &j);

printf (“%d %d \ n”, i, j);

return 0;

}

GATE CSE 2010

1. 2 2

2. 2 1

3. 0 1

4. 0 2

Solution:

The correct option is 4

6)The output of the following C program is

GATE CSE 2018

#include< stdio.h >

struct Ournode{

char x,y,z;

};

int main(){

struct Ournode p = {‘1’, ‘0’, ‘a’+2};

struct Ournode \*q = &p;

printf (“%c, %c”, \*((char\*)q+1), \*((char\*)q+2));

return 0;

}

1. 0, c

2. 0, a+2

3. ‘0’, ‘a+2’

4. ‘0’,’c’

Output: 0,a+2

7)The output of the following C program is

#include < stdio.h >

void mystery(int \*ptra, int \*ptrb) {

int \*temp;

temp = ptrb;

ptrb = ptra;

ptra = temp;

}

int main() {

int a=2016, b=0, c=4, d=42;

mystery(&a, &b);

if (a < c)

mystery(&c, &a);

mystery(&a, &d);

printf(“%d\n”, a);

}

Output: 2016

8)The output of the following C program is

#include < stdio.h >

int main () {

int arr [] = {1,2,3,4,5,6,7,8,9,0,1,2,5}, \*ip = arr+4;

printf (“%d\n”, ip[1]);

return 0;

}

Output: 6

9)Consider the following C function

 GATE CSE 2004

void swap (int a, int b)

{

int temp;

temp = a;

a = b;

b = temp;

}

In order to exchange the values of two variables x and y

1. call swap (x,y)

2. call swap (&x, &y)

3. swap (x, y) cannot be used as it does return any value

4. swap (x,y) cannot be used as the parameters are passed by value

Solution:

The correct option is 2

In case of eexchaning values of 2 variables when we pass the values of variables in function the changes will not reflect in the main function but when we pass the address of the variable then the changes will reflect in main function the values will exchange. This is called call by reference .