

Programming Techniques

ICT 131-3

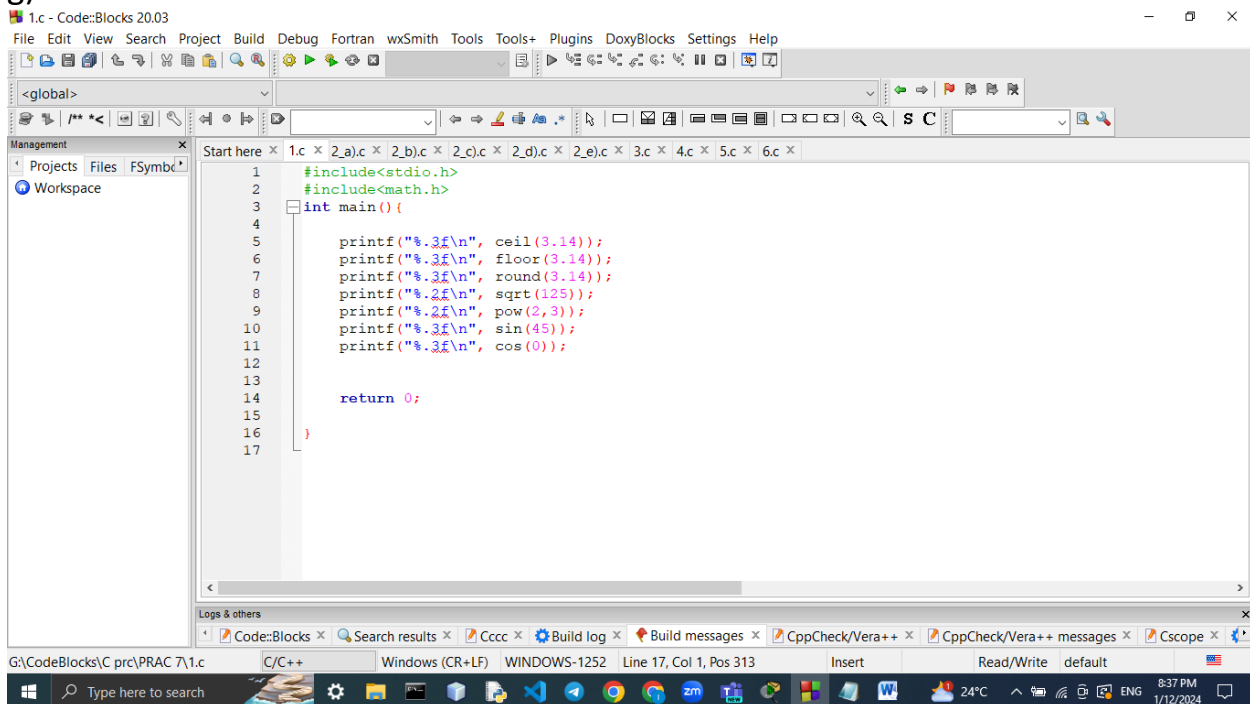
Practical 07

UWU/ICT/22/077

H.M.S.H.HERATH

1. Write a simple C program using maths functions to give the output for below cases:

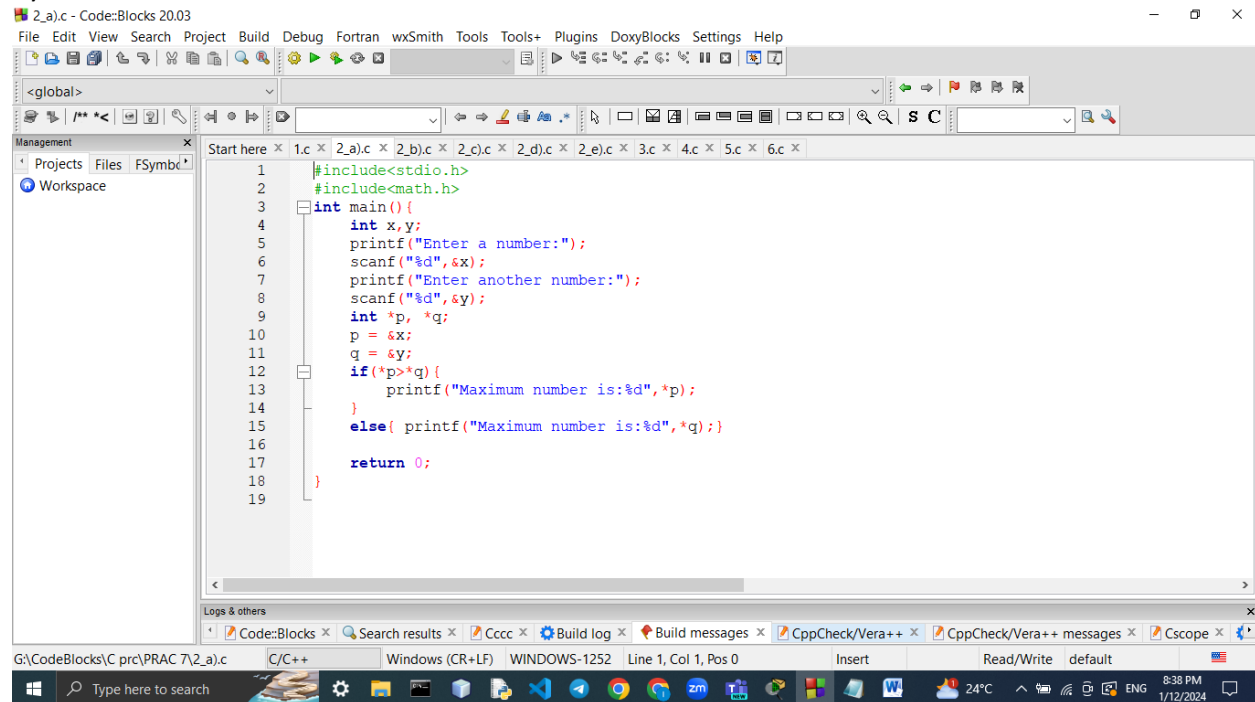
- a) round up
- b) round down
- c) round off to the nearest value
- d) square root
- e) power of the given two arguments
- f) sin value
- g) cos value



```
1  #include<stdio.h>
2  #include<math.h>
3  int main() {
4
5      printf("%.3f\n", ceil(3.14));
6      printf("%.3f\n", floor(3.14));
7      printf("%.3f\n", round(3.14));
8      printf("%.2f\n", sqrt(125));
9      printf("%.2f\n", pow(2,3));
10     printf("%.3f\n", sin(45));
11     printf("%.3f\n", cos(0));
12
13
14     return 0;
15
16
17 }
```

2. Write C programs using pointers

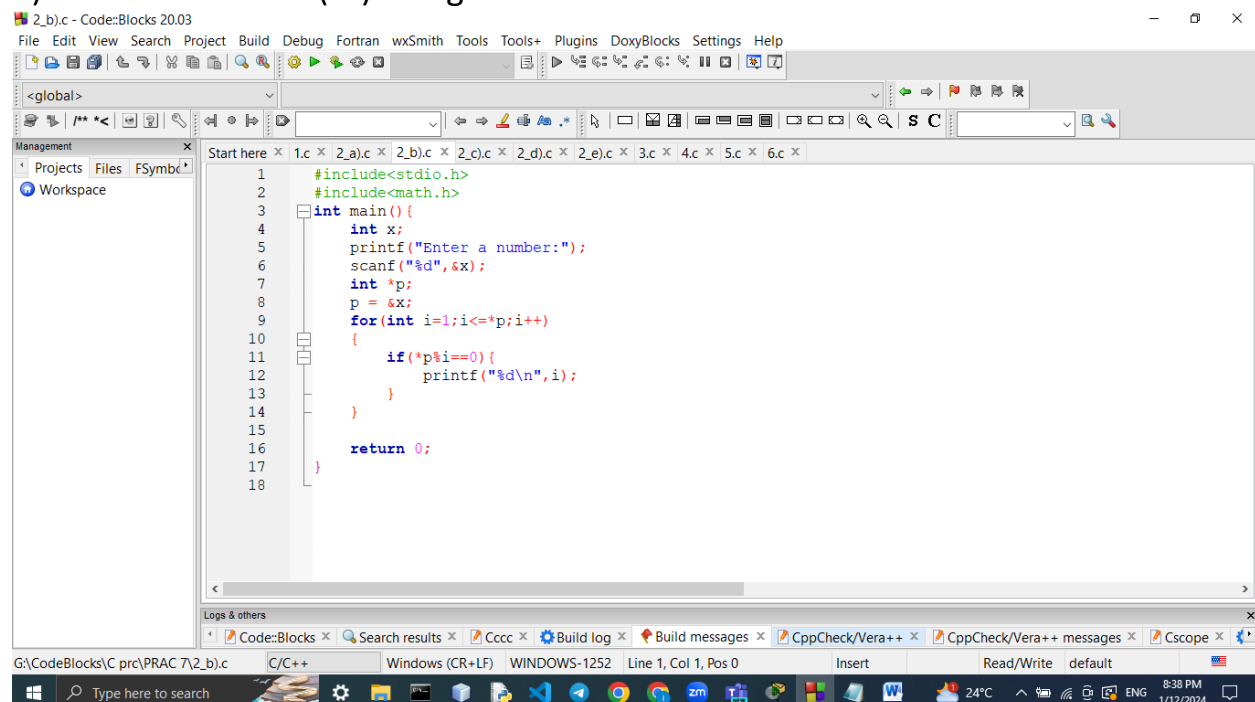
a) Find the maximum number between two numbers



The screenshot shows the Code::Blocks IDE with a C program named 2_a.c. The program includes `<stdio.h>` and `<math.h>`. It defines a `main` function that takes two integers `x` and `y` as input. It then declares two pointers `p` and `q` and assigns them the addresses of `x` and `y` respectively. A conditional statement checks if the value at `*p` is greater than the value at `*q`. If true, it prints the maximum value from `*p`; otherwise, it prints the maximum value from `*q`. The program returns 0.

```
1 #include<stdio.h>
2 #include<math.h>
3 int main(){
4     int x,y;
5     printf("Enter a number:");
6     scanf("%d",&x);
7     printf("Enter another number:");
8     scanf("%d",&y);
9     int *p, *q;
10    p = &x;
11    q = &y;
12    if(*p>*q){
13        printf("Maximum number is:%d",*p);
14    }
15    else{ printf("Maximum number is:%d",*q);}
16
17    return 0;
18 }
19
```

b) Find the factorial (x!) of a given number



The screenshot shows the Code::Blocks IDE with a C program named 2_b.c. The program includes `<stdio.h>` and `<math.h>`. It defines a `main` function that takes an integer `x` as input. It then declares a pointer `p` and assigns it the address of `x`. A `for` loop iterates from 1 to the value at `*p`. Inside the loop, it checks if the value at `*p` is zero. If true, it prints the factorial value from `*p`. The program returns 0.

```
1 #include<stdio.h>
2 #include<math.h>
3 int main(){
4     int x;
5     printf("Enter a number:");
6     scanf("%d",&x);
7     int *p;
8     p = &x;
9     for(int i=1;i<=*p;i++)
10    {
11        if(*p%i==0){
12            printf("%d\n",i);
13        }
14    }
15
16    return 0;
17 }
18
```

c) Compute the sum of all elements in an array

The screenshot shows the Code::Blocks IDE with a C program open in the editor. The program calculates the sum of elements in an array. The code is as follows:

```
1 #include<stdio.h>
2 #include<math.h>
3
4 int main() {
5     int array[5]={5,4,8,9,2};
6     int sum;
7     for(int i=0;i<5;i++)
8     {
9         sum+=array[i];
10    }
11    printf("Sum of all the elements in the array:%d\n",sum);
12
13    return 0;
14 }
```

The IDE interface includes a menu bar, a toolbar, a project manager on the left, and a status bar at the bottom showing the file path and line/col information.

d) Print the elements of an array in reverse order

The screenshot shows the Code::Blocks IDE with a C program open in the editor. The program prints the elements of an array in reverse order. The code is as follows:

```
1 #include<stdio.h>
2 int main(){
3     int n;
4     int rev[5],array[5]={5,4,8,9,2};
5     int a=4,b=0;
6     while(a>=0){
7         rev[b]=array[a];
8         a--;
9         b++;
10    }
11    printf("Reverse of the array:\n");
12    for (int k = 0; k < 5; k++) {
13        printf("%d,",rev[k]);
14    }
15    return 0;
16 }
```

The IDE interface is similar to the previous screenshot, showing the same menu bar, toolbar, and project manager.

e) Search an element in an array .

2_e).c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main(): int

```
1 #include<stdio.h>
2
3 int main(){
4     int n,x;
5     int array[n];
6     printf("Enter size for the array:");
7     scanf("%d",&n);
8     for (int i = 0; i < n; i++ ) {
9         printf("Enter values for the array:");
10        scanf("%d",&array[i]);
11    }
12    printf("Search the element:");
13    scanf("%d",&x);
14    printf("%d",array[x]);
15
16    return 0;
17 }
18
```

Management

Projects Files FSymbc

Workspace

Start here x 1.c x 2_a).c x 2_b).c x 2_c).c x 2_d).c x 2_e).c x 3.c x 4.c x 5.c x 6.c x

Logs & others

Code::Blocks x Search results x Cccc x Build log x Build messages x CppCheck/Vera++ x CppCheck/Vera++ messages x Cscope x

G:\CodeBlocks\C prc\PRAC 7\2_e).c C/C++ Windows (CR+LF) WINDOWS-1252 Line 4, Col 18, Pos 63 Insert Read/Write default ENG 8:39 PM 1/12/2024

3. Write a C program to find largest number in an array

3.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

```
1 #include<stdio.h>
2
3 int main(){
4     int i=0,n,max=0;
5     int array[n];
6     printf("Enter size for the array:");
7     scanf("%d",&n);
8     for (int i = 0; i < n; i++ ) {
9         printf("Enter values for the array:");
10        scanf("%d",&array[i]);
11    }
12    while(i<n){
13        if(array[i]>max) max=array[i];
14        else max=max;
15        i++;
16    }
17    printf("Largest number is:%d",max);
18
19    return 0;
20 }
```

Management

Projects Files FSymbc

Workspace

Start here x 1.c x 2_a).c x 2_b).c x 2_c).c x 2_d).c x 2_e).c x 3.c x 4.c x 5.c x 6.c x

Logs & others

Code::Blocks x Search results x Cccc x Build log x Build messages x CppCheck/Vera++ x CppCheck/Vera++ messages x Cscope x

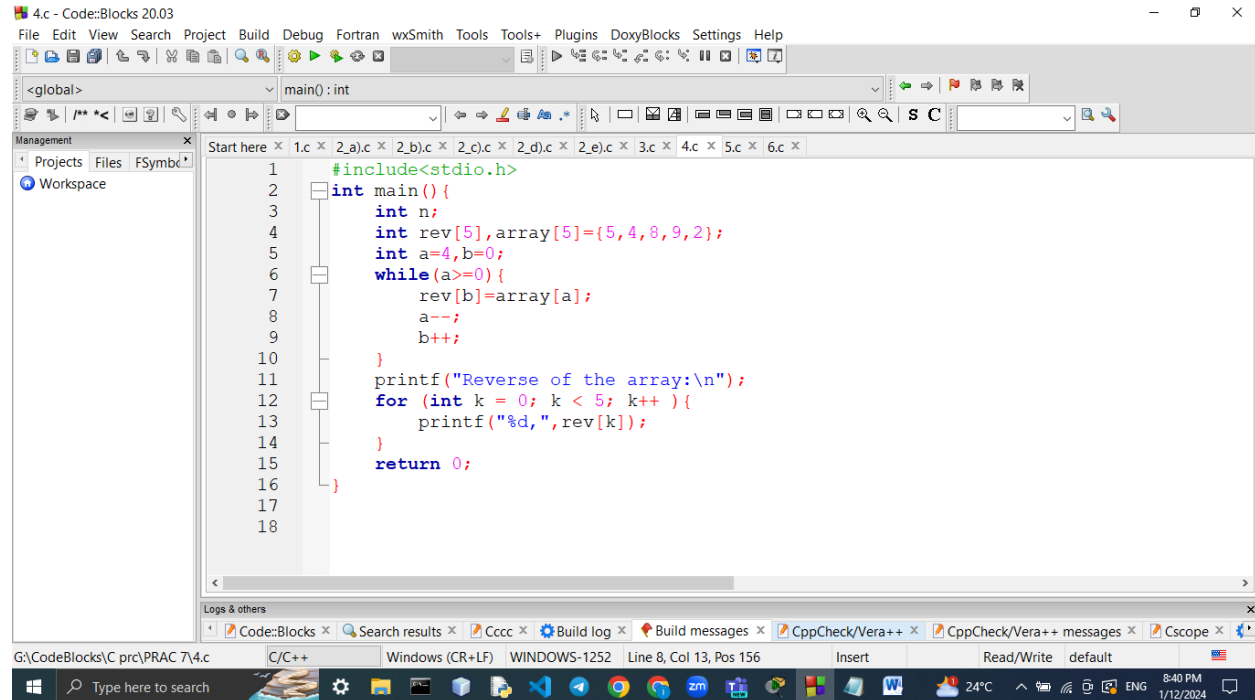
G:\CodeBlocks\C prc\PRAC 7\3.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert Read/Write default ENG 8:40 PM 1/12/2024

4. Write a C program to find reverse of an array

Example: If the elements of the array are: 10, 5, 16, 35, 500

Then its reverse would be: 500, 35, 16, 5, 10

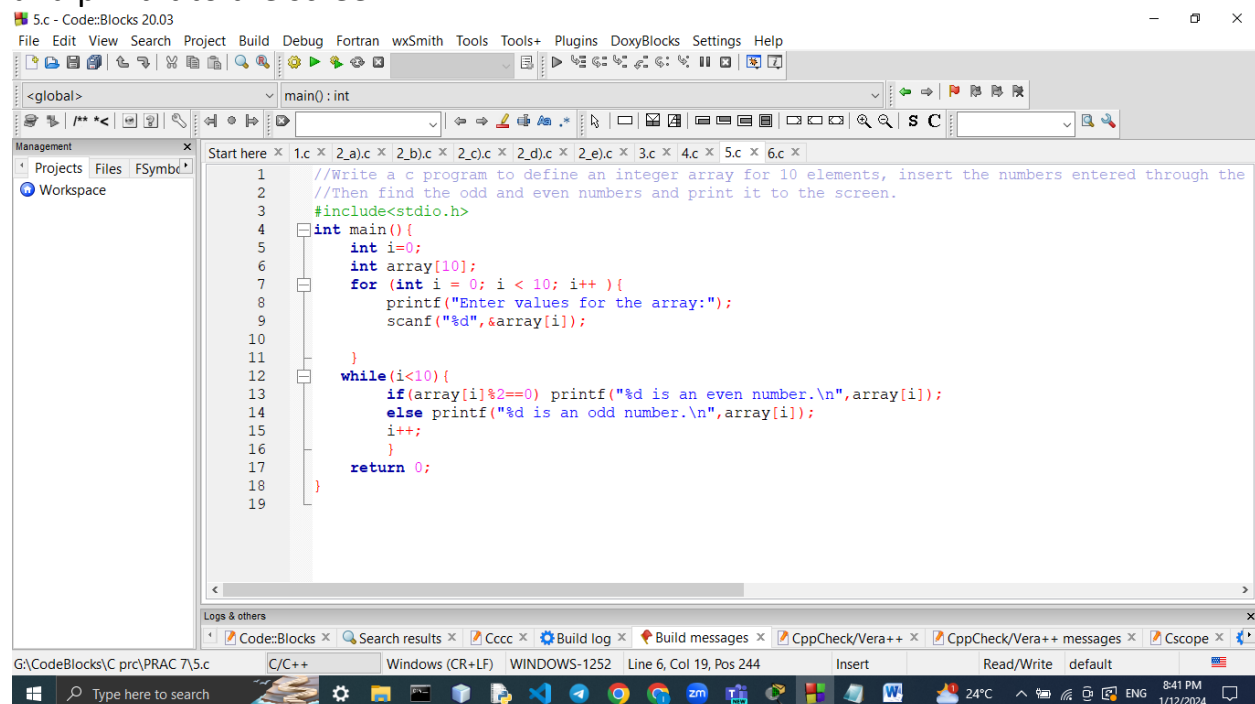
Hint: Can Use 2 arrays



The screenshot shows the Code::Blocks IDE with a C program to reverse an array. The program uses two arrays: 'array' and 'rev'. It reads 5 elements into 'array' and then copies them in reverse order into 'rev'. The output is 'Reverse of the array:\n' followed by the reversed elements: 500, 35, 16, 5, 10.

```
1 #include<stdio.h>
2 int main() {
3     int n;
4     int rev[5], array[5]={5,4,8,9,2};
5     int a=4,b=0;
6     while(a>=0) {
7         rev[b]=array[a];
8         a--;
9         b++;
10    }
11    printf("Reverse of the array:\n");
12    for (int k = 0; k < 5; k++) {
13        printf("%d", rev[k]);
14    }
15    return 0;
16 }
17
18
```

5. Write a c program to define an integer array for 10 elements, insert the numbers entered through the keyboard. Then find the odd and even numbers and print it to the screen.



The screenshot shows the Code::Blocks IDE with a C program to check odd and even numbers in an array. The program defines an array of 10 elements, reads 10 numbers from the keyboard, and then iterates through the array to print whether each number is even or odd.

```
1 //Write a c program to define an integer array for 10 elements, insert the numbers entered through the
2 //Then find the odd and even numbers and print it to the screen.
3 #include<stdio.h>
4 int main() {
5     int i=0;
6     int array[10];
7     for (int i = 0; i < 10; i++) {
8         printf("Enter values for the array:");
9         scanf("%d",&array[i]);
10    }
11    while(i<10){
12        if(array[i]%2==0) printf("%d is an even number.\n",array[i]);
13        else printf("%d is an odd number.\n",array[i]);
14        i++;
15    }
16    return 0;
17 }
18
19
```

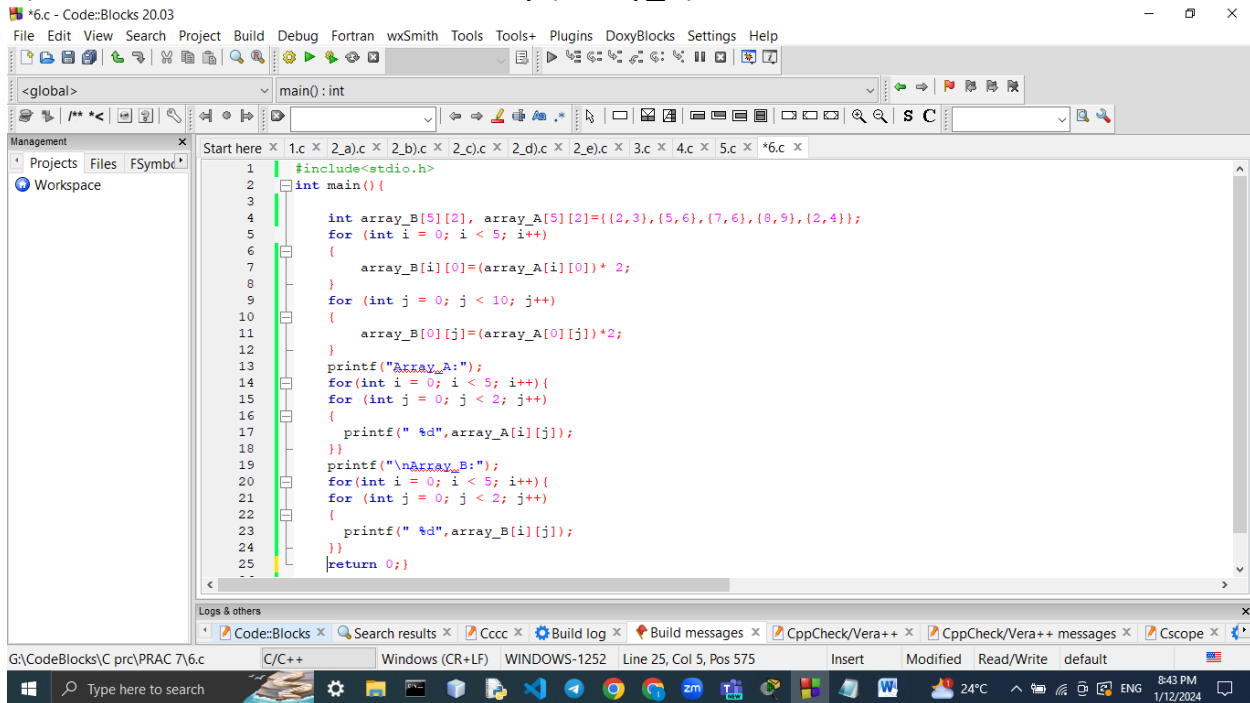
6. Write a C program to obtain given array(array_B) from the following array(array_A)

array_A={{2,3},{5,6},{7,6},{8,9},{2,4}};

array_B = {{4,6},{10,12},{14,12},{16,18},{4,8}};

a) Print the values in the array(array_A)

b) Print the values in the new array (array_B)



The screenshot shows the Code::Blocks IDE with a C program open. The program defines two arrays, array_A and array_B, and prints their values. The code is as follows:

```
1 #include<stdio.h>
2 int main() {
3
4     int array_B[5][2], array_A[5][2]={{2,3},{5,6},{7,6},{8,9},{2,4}};
5     for (int i = 0; i < 5; i++)
6     {
7         array_B[i][0]=(array_A[i][0]) * 2;
8     }
9     for (int j = 0; j < 10; j++)
10    {
11        array_B[0][j]=(array_A[0][j]) * 2;
12    }
13    printf("ARRAY_A:");
14    for(int i = 0; i < 5; i++){
15        for (int j = 0; j < 2; j++)
16        {
17            printf(" %d",array_A[i][j]);
18        }
19        printf("\nARRAY_B:");
20        for(int i = 0; i < 5; i++){
21            for (int j = 0; j < 2; j++)
22            {
23                printf(" %d",array_B[i][j]);
24            }
25        }
26    }
27    return 0;
28 }
```

The IDE interface includes a menu bar (File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help), a toolbar, a project manager on the left, and a status bar at the bottom showing the file path G:\CodeBlocks\C prc\PRAC 7\6.c, the current language C/C++, and the system clock 8:43 PM 1/12/2024.