



Practical 05



PS Gunathilake
30138

Practical 05

Section A

1)

<ul style="list-style-type: none">• While	<ul style="list-style-type: none">• DO While
<pre>#include <stdio.h> int main() { int number = 0; while (number <= 100) { printf("%d ", number); number++; } return 0; }</pre>	<pre>#include <stdio.h> int main() { int number = 0; do { printf("%d ", number); number++; } while (number <= 100); return 0; }</pre>

<ul style="list-style-type: none">• For
<pre>#include <stdio.h> int main() { for (int number = 0; number <= 100; number++) { printf("%d ", number); } return 0; }</pre>

```
1 #include <stdio.h>
2
3 int main() {
4     int number = 0;
5
6     while (number <= 100) {
7         printf("%d ", number);
8         number++;
9     }
10
11     return 0;
12 }
13
14
```

```
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 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 6
7 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88
89 90 91 92 93 94 95 96 97 98 99 100
Process returned 0 (0x0)   execution time : 0.051 s
Press any key to continue.
```

While

```
1 #include <stdio.h>
2
3 int main() {
4     int number = 0;
5
6     do {
7         printf("%d ", number);
8         number++;
9     } while (number <= 100);
10
11     return 0;
12 }
13
14
```

```
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 50 51 5
2 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68
69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84
85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
Process returned 0 (0x0)   execution time : 0.049
s
Press any key to continue.
```

Do While

```
1 #include <stdio.h>
2
3 int main() {
4     for (int number = 0; number <= 100; number++) {
5         printf("%d ", number);
6     }
7
8     return 0;
9 }
10
11
12
```

```
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 4
7 48 49 50 51 52 53 54 55 56 57 58 59 60 61
62 63 64 65 66 67 68 69 70 71 72 73 74 75 76
77 78 79 80 81 82 83 84 85 86 87 88 89 90 9
1 92 93 94 95 96 97 98 99 100
Process returned 0 (0x0)   execution time :
0.049 s
Press any key to continue.
```

For

2)

```
#include <stdio.h>

int main() {

    int marks[10];

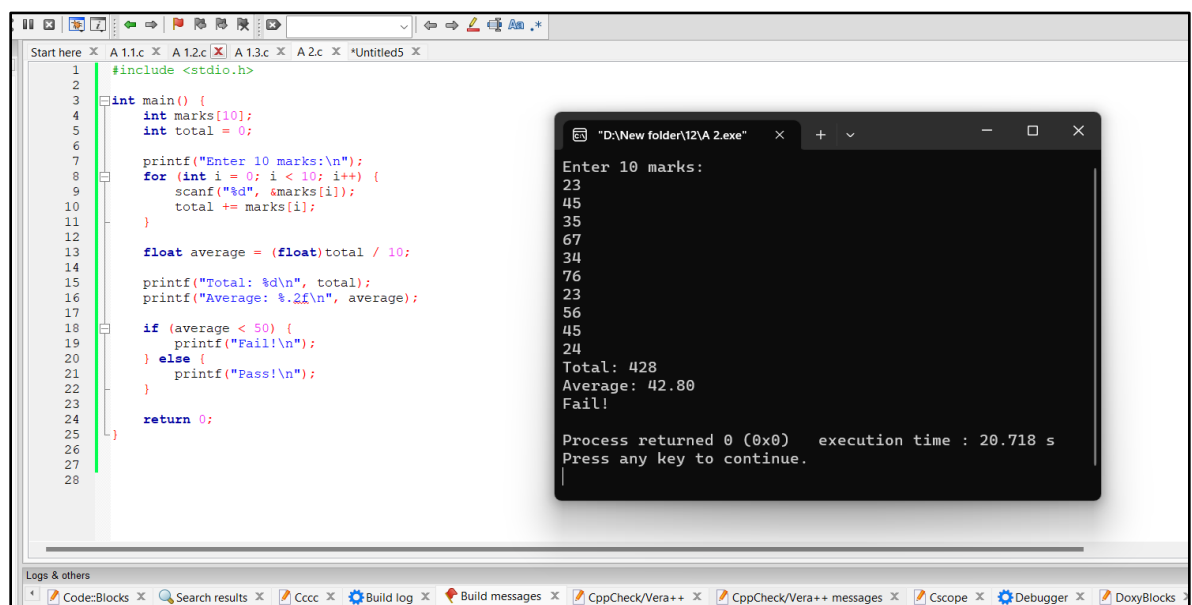
    int total = 0;


    printf("Enter 10 marks:\n");
    for (int i = 0; i < 10; i++) {
        scanf("%d", &marks[i]);
        total += marks[i];
    }

    float average = (float)total / 10;
    printf("Total: %d\n", total);
    printf("Average: %.2f\n", average);

    if (average < 50) {
        printf("Fail!\n");
    } else {
        printf("Pass!\n");
    }

    return 0;
}
```



The screenshot shows a C++ IDE with the code from the previous block. The code is compiled and executed. The output window shows the following text:

```
Enter 10 marks:
23
45
35
67
34
76
23
56
45
24
Total: 428
Average: 42.80
Fail!

Process returned 0 (0x0)   execution time : 20.718 s
Press any key to continue.
```

3)

```
#include <stdio.h>

int main() {

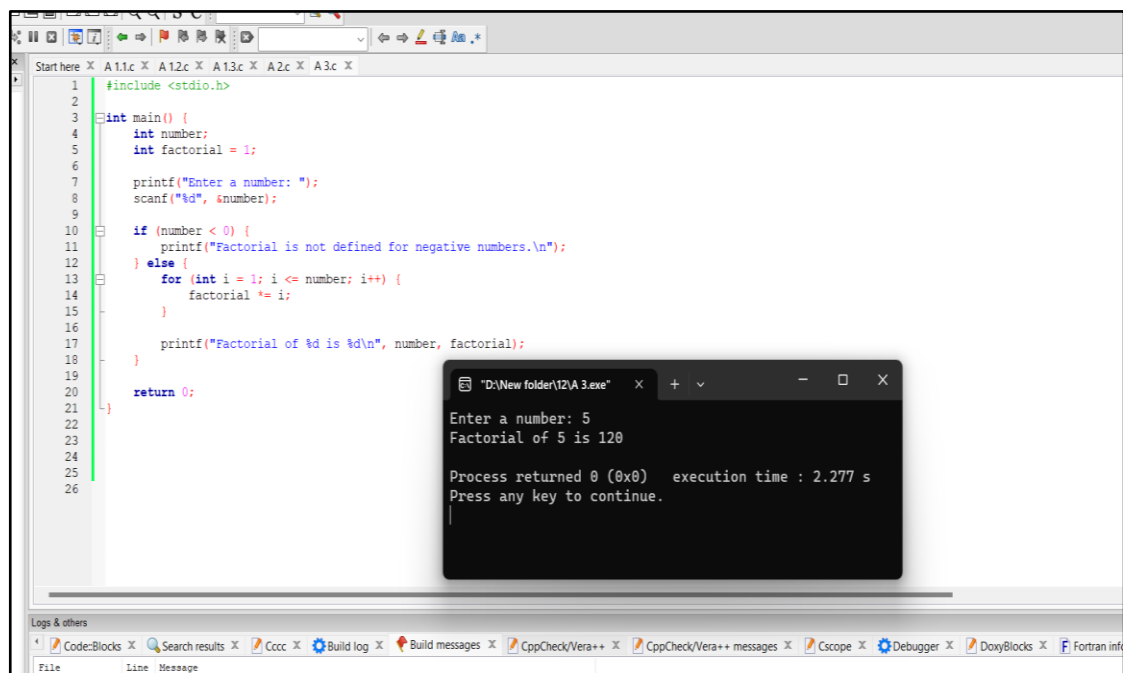
    int number;

    int factorial = 1;

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

    if (number < 0) {
        printf("Factorial is not defined for negative numbers.\n");
    } else {
        for (int i = 1; i <= number; i++) {
            factorial *= i;
        }
        printf("Factorial of %d is %d\n", number, factorial);
    }

    return 0;
}
```



The screenshot shows a code editor with the following C program code for calculating factorial:

```
1 #include <stdio.h>
2
3 int main() {
4     int number;
5     int factorial = 1;
6
7     printf("Enter a number: ");
8     scanf("%d", &number);
9
10    if (number < 0) {
11        printf("Factorial is not defined for negative numbers.\n");
12    } else {
13        for (int i = 1; i <= number; i++) {
14            factorial *= i;
15        }
16        printf("Factorial of %d is %d\n", number, factorial);
17    }
18
19    return 0;
20 }
```

Below the code editor, a terminal window titled "D:\New folder\12\A 3.exe" displays the program's execution:

```
Enter a number: 5
Factorial of 5 is 120

Process returned 0 (0x0)   execution time : 2.277 s
Press any key to continue.
```

4)

```
#include <stdio.h>

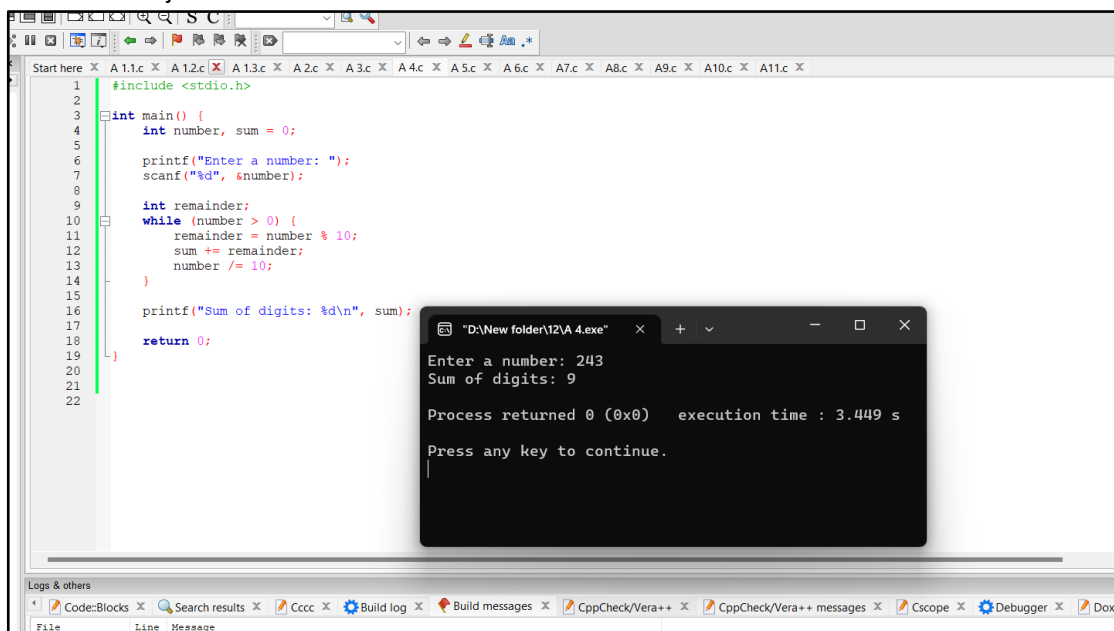
int main() {
    int number, sum = 0;

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

    int remainder;
    while (number > 0) {
        remainder = number % 10;
        sum += remainder;
        number /= 10;
    }

    printf("Sum of digits: %d\n", sum);

    return 0;
}
```



The screenshot shows a C++ IDE with the code from the previous block. The code is written in a file named 'A11.c'. The code calculates the sum of digits of a number. The output window shows the execution results for the input '243'.

```
1 #include <stdio.h>
2
3
4 int main() {
5     int number, sum = 0;
6
7     printf("Enter a number: ");
8     scanf("%d", &number);
9
10    int remainder;
11    while (number > 0) {
12        remainder = number % 10;
13        sum += remainder;
14        number /= 10;
15    }
16
17    printf("Sum of digits: %d\n", sum);
18
19    return 0;
20 }
21
22
```

Output window: "D:\New folder\12\A 4.exe"

```
Enter a number: 243
Sum of digits: 9

Process returned 0 (0x0)   execution time : 3.449 s
Press any key to continue.
```

5)

```
#include <stdio.h>

int main() {

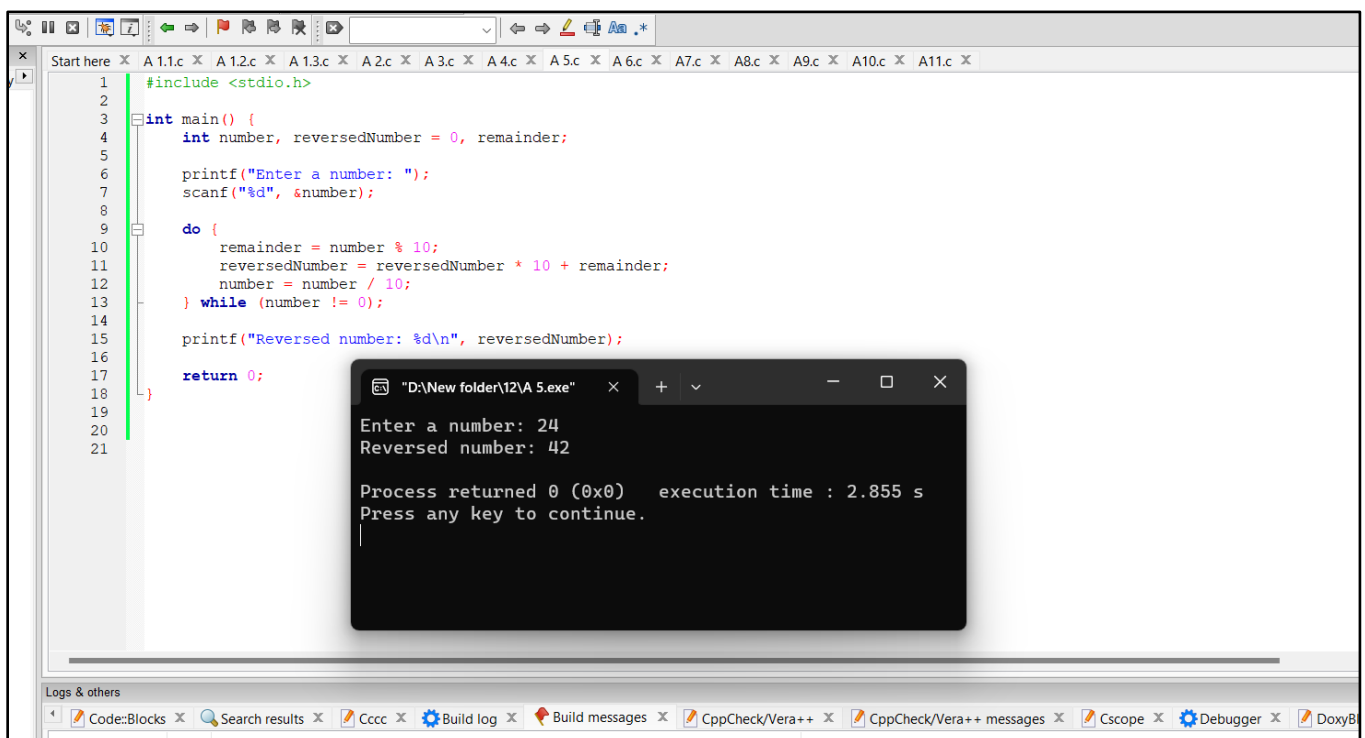
    int number, reversedNumber = 0, remainder;

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

    do {
        remainder = number % 10;
        reversedNumber = reversedNumber * 10 + remainder;
        number = number / 10;
    } while (number != 0);

    printf("Reversed number: %d\n", reversedNumber);

    return 0;
}
```



The screenshot shows a code editor with the C program from the previous block. The code is written in a file named 'A 5.c'. The editor has a toolbar at the top and a sidebar on the left. A terminal window is open in the foreground, showing the execution of the program. The terminal output is as follows:

```
"D:\New folder\12\A 5.exe" x + - □ ×
Enter a number: 24
Reversed number: 42

Process returned 0 (0x0)   execution time : 2.855 s
Press any key to continue.
```

6)

```
#include <stdio.h>

int main() {

    int base, exponent, result = 1;

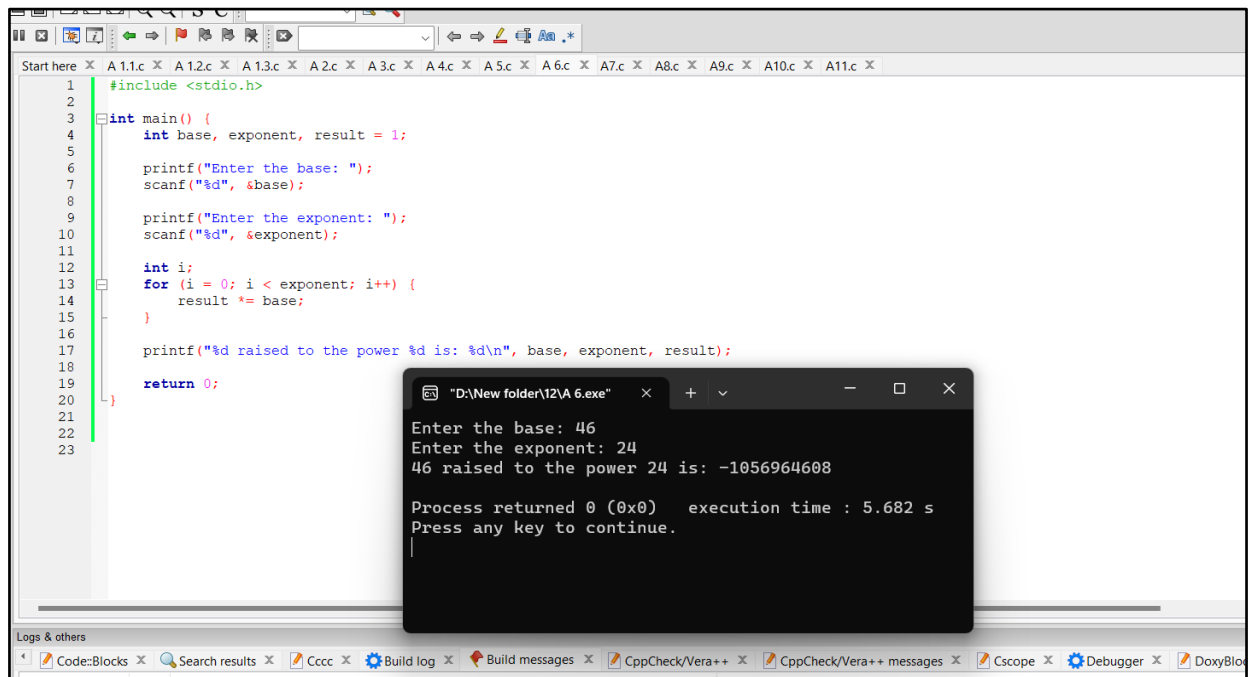

    printf("Enter the base: ");
    scanf("%d", &base);


    printf("Enter the exponent: ");
    scanf("%d", &exponent);


    int i;
    for (i = 0; i < exponent; i++) {
        result *= base;
    }

    printf("%d raised to the power %d is: %d\n", base, exponent, result);

    return 0;
}
```



The screenshot shows a code editor with the C program from the previous block. The code is color-coded and includes line numbers. A terminal window is open in the foreground, displaying the program's execution. The user has entered '46' for the base and '24' for the exponent. The program outputs '-1056964608' as the result of 46 raised to the power of 24. The terminal also shows the process return code as 0 and the execution time as 5.682 seconds.

```
Start here x A 1.1.c x A 1.2.c x A 1.3.c x A 2.c x A 3.c x A 4.c x A 5.c x A 6.c x A 7.c x A 8.c x A 9.c x A 10.c x A 11.c x
1 #include <stdio.h>
2
3 int main() {
4     int base, exponent, result = 1;
5
6     printf("Enter the base: ");
7     scanf("%d", &base);
8
9     printf("Enter the exponent: ");
10    scanf("%d", &exponent);
11
12    int i;
13    for (i = 0; i < exponent; i++) {
14        result *= base;
15    }
16
17    printf("%d raised to the power %d is: %d\n", base, exponent, result);
18
19    return 0;
20 }
21
22
23
```

```
"D:\New folder\12\A 6.exe" x + v - □ ×
Enter the base: 46
Enter the exponent: 24
46 raised to the power 24 is: -1056964608

Process returned 0 (0x0)   execution time : 5.682 s
Press any key to continue.
|
```

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 Debugger x DoxyBlo

7)

```
#include <stdio.h>
```

```
int main() {
```

```
    int n = 10;
```

```
    int fib[n];
```

```
    int i;
```

```
    fib[0] = 0;
```

```
    fib[1] = 1;
```

```
    for (i = 2; i < n; i++) {
```

```
        fib[i] = fib[i-1] + fib[i-2];
```

```
    }
```

```
    printf("The first 10 numbers of the Fibonacci sequence are:\n");
```

```
    for (i = 0; i < n; i++) {
```

```
        printf("%d ", fib[i]);
```

```
    }
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

The screenshot shows a code editor with the following C code for calculating the first 10 Fibonacci numbers:

```
1 #include <stdio.h>
2
3 int main() {
4     int n = 10;
5     int fib[n];
6     int i;
7
8     fib[0] = 0;
9     fib[1] = 1;
10
11     for (i = 2; i < n; i++) {
12         fib[i] = fib[i-1] + fib[i-2];
13     }
14
15     printf("The first 10 numbers of the Fibonacci sequence are:\n");
16     for (i = 0; i < n; i++) {
17         printf("%d ", fib[i]);
18     }
19     printf("\n");
20
21     return 0;
22 }
23
24
25
```

Below the code editor, a terminal window titled "D:\New folder\12\A7.exe" displays the output of the program:

```
The first 10 numbers of the Fibonacci sequence are:
0 1 1 2 3 5 8 13 21 34

Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

The bottom of the screenshot shows the IDE's status bar with various tool windows open, including "Logs & others", "Code:Blocks", "Search results", "Cccc", "Build log", "Build messages", "CppCheck/Vera++", "CppCheck/Vera++ messages", "Cscope", "Debugger", "DoxyBlocks", and "Fortra".

8)

```
#include <stdio.h>

int main() {
    int number, originalNumber, remainder, result = 0, n = 0;

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

    originalNumber = number;

    while (originalNumber != 0) {
        originalNumber /= 10;
        ++n;
    }

    originalNumber = number;

    while (originalNumber != 0) {
        remainder = originalNumber % 10;
        int power = 1;
        for (int i = 1; i <= n; ++i) {
            power *= remainder;
        }
        result += power;
        originalNumber /= 10;
    }

    if (result == number)
        printf("%d is an Armstrong number.\n", number);
}
```

```

else

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

return 0;
}

```

The screenshot shows a code editor with the following C code:

```

1 #include <stdio.h>
2
3 int main() {
4     int number, originalNumber, remainder, result = 0, n = 0;
5
6     printf("Enter a number: ");
7     scanf("%d", &number);
8
9     originalNumber = number;
10
11     while (originalNumber != 0) {
12         originalNumber /= 10;
13         ++n;
14     }
15
16     originalNumber = number;
17
18     while (originalNumber != 0) {
19         remainder = originalNumber % 10;
20         int power = 1;
21         for (int i = 1; i <= n; ++i) {
22             power *= remainder;
23         }
24         result += power;
25         originalNumber /= 10;
26     }
27
28     if (result == number)
29         printf("%d is an Armstrong number.\n", number);
30     else
31         printf("%d is not an Armstrong number.\n", number);
32 }

```

Overlaid on the code editor is a terminal window titled "D:\New folder\12\A8.exe" showing the execution output:

```

Enter a number: 12
12 is not an Armstrong number.

Process returned 0 (0x0)   execution time : 3.287 s
Press any key to continue.

```

9)

```

#include <stdio.h>

int main() {
    char letter;

    printf("ASCII values for letters A to Z:\n");

    for (letter = 'A'; letter <= 'Z'; ++letter) {
        printf("%c: %d\n", letter, letter);
    }
}

```

```

    }

    return 0;
}

```

The screenshot shows a code editor with the following C code:

```

1  #include <stdio.h>
2
3  int main() {
4      char letter;
5
6      printf("ASCII values for letters A to Z:\n");
7
8      for (letter = 'A'; letter <= 'Z'; ++letter) {
9          printf("%c: %d\n", letter, letter);
10     }
11
12     return 0;
13 }
14
15
16

```

Overlaid on the editor is a terminal window titled "D:\New folder\12\A9.exe" showing the output of the program:

```

A: 65
B: 66
C: 67
D: 68
E: 69
F: 70
G: 71
H: 72
I: 73
J: 74
K: 75
L: 76
M: 77
N: 78
O: 79
P: 80
Q: 81
R: 82
S: 83
T: 84
U: 85
V: 86
W: 87
X: 88
Y: 89
Z: 90

```

At the bottom of the terminal window, it says: "Process returned 0 (0x0) execution time : 0.045 s Press any key to continue."

10)

```

#include <stdio.h>

int main() {
    int rows = 5; // number of rows in the pattern
    int i, j;

    for (i = 1; i <= rows; ++i) {
        for (j = 1; j <= i; ++j) {
            printf("*");
        }
    }
}

```

```

        printf("\n");
    }

    return 0;
}

```

```

1  #include <stdio.h>
2
3  int main() {
4      int rows = 5; // number of rows in the pattern
5      int i, j;
6
7      for (i = 1; i <= rows; ++i) {
8          for (j = 1; j <= i; ++j) {
9              printf("*");
10             }
11             printf("\n");
12         }
13
14         return 0;
15     }
16
17
18

```

Output in console:

```

*
**
***
****
*****

Process returned 0 (0x0)   execution time : 0.116 s
Press any key to continue.

```

11)

```

#include <stdio.h>

int main() {
    int number, i, isPrime = 1;

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

    if (number == 0 || number == 1) {
        isPrime = 0;
    }
}

```

```

    } else {
        for (i = 2; i <= number / 2; ++i) {
            if (number % i == 0) {
                isPrime = 0;
                break;
            }
        }
    }

    if (isPrime) {
        printf("%d is a prime number.\n", number);
    } else {
        printf("%d is not a prime number.\n", number);
    }

    return 0;
}

```

The screenshot shows a code editor with a C program that checks if a number is prime. The code is as follows:

```

1  #include <stdio.h>
2
3  int main() {
4      int number, i, isPrime = 1;
5
6      printf("Enter a positive integer: ");
7      scanf("%d", &number);
8
9      if (number == 0 || number == 1) {
10         isPrime = 0;
11     } else {
12         for (i = 2; i <= number / 2; ++i) {
13             if (number % i == 0) {
14                 isPrime = 0;
15                 break;
16             }
17         }
18     }
19
20     if (isPrime) {
21         printf("%d is a prime number.\n", number);
22     } else {
23         printf("%d is not a prime number.\n", number);
24     }
25
26     return 0;
27 }
28
29

```

The output window shows the execution results for the input 25:

```

D:\New folder\12\A11.exe
Enter a positive integer: 25
25 is not a prime number.

Process returned 0 (0x0)   execution time : 5.348 s
Press any key to continue.

```

The bottom of the screenshot shows the IDE's status bar with various tool windows like 'Logs & others', 'Code::Blocks', 'Search results', 'Cccc', 'Build log', 'Build messages', 'CppCheck/Vera++', 'CppCheck/Vera++ messages', 'Cscope', 'Debugger', and 'DoxyBlocks'.

12)

```
#include <stdio.h>

int main() {

    int number, i;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    printf("Factors of %d are: ", number);

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

        if (number % i == 0) {

            printf("%d ", i);

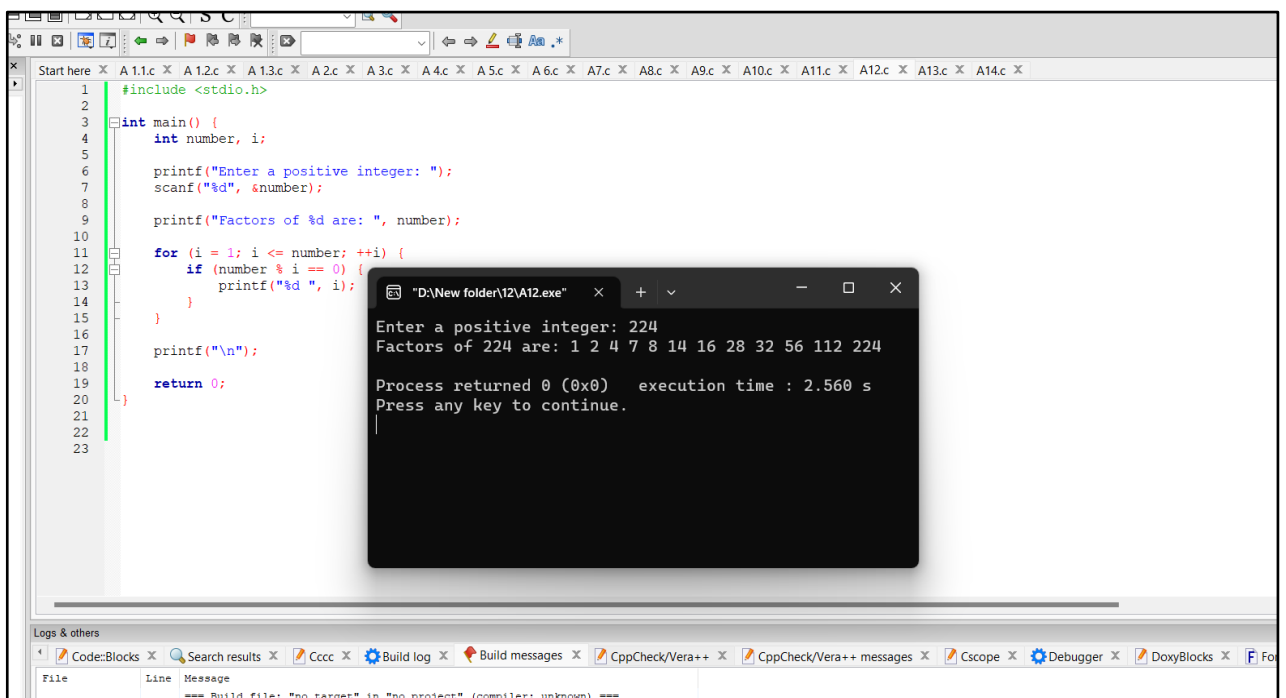
        }

    }

    printf("\n");

    return 0;

}
```



The screenshot shows a C++ IDE with a code editor and a console window. The code in the editor is the same as the one above. The console window shows the output of the program when it is run. The user enters 224, and the program outputs the factors of 224: 1 2 4 7 8 14 16 28 32 56 112 224. The console also shows the process returned 0 (0x0) and the execution time was 2.560 s. The console prompt is 'Press any key to continue.' The IDE interface includes a toolbar at the top, a file explorer on the left, and a status bar at the bottom.

```
1 #include <stdio.h>
2
3 int main() {
4     int number, i;
5
6     printf("Enter a positive integer: ");
7     scanf("%d", &number);
8
9     printf("Factors of %d are: ", number);
10
11     for (i = 1; i <= number; ++i) {
12         if (number % i == 0) {
13             printf("%d ", i);
14         }
15     }
16
17     printf("\n");
18
19     return 0;
20 }
21
22
23
```

Enter a positive integer: 224
Factors of 224 are: 1 2 4 7 8 14 16 28 32 56 112 224
Process returned 0 (0x0) execution time : 2.560 s
Press any key to continue.

13)

```
#include <stdio.h>

int main() {
    int number;
    int sum = 0;

    printf("Enter numbers to be added (enter -1 to stop):\n");

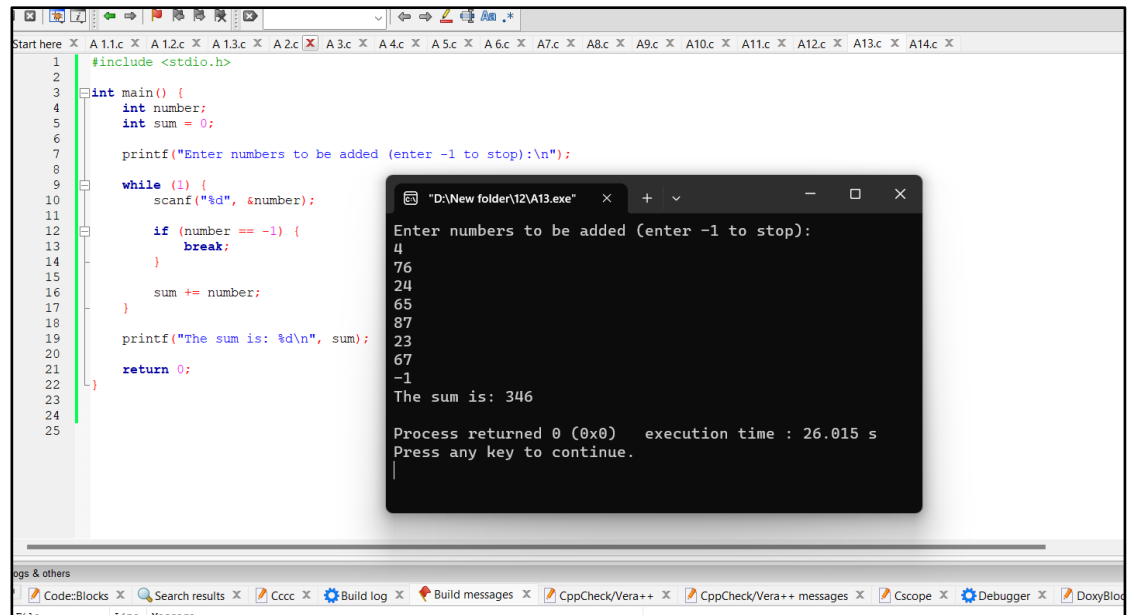
    while (1) {
        scanf("%d", &number);

        if (number == -1) {
            break;
        }

        sum += number;
    }

    printf("The sum is: %d\n", sum);

    return 0;
}
```



The screenshot displays a code editor with a C program for calculating the sum of numbers. The code is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int number;
5     int sum = 0;
6
7     printf("Enter numbers to be added (enter -1 to stop):\n");
8
9     while (1) {
10         scanf("%d", &number);
11
12         if (number == -1) {
13             break;
14         }
15
16         sum += number;
17     }
18
19     printf("The sum is: %d\n", sum);
20
21     return 0;
22 }
23
24
25
```

Overlaid on the code editor is a terminal window titled "D:\New folder\12\A13.exe". The terminal shows the program's execution:

```
Enter numbers to be added (enter -1 to stop):
4
76
24
65
87
23
67
-1
The sum is: 346

Process returned 0 (0x0)   execution time : 26.015 s
Press any key to continue.
```

The bottom of the screenshot shows the taskbar with various open applications, including Code::Blocks, Search results, Cccc, Build log, Build messages, CppCheck/Vera++, CppCheck/Vera++ messages, Cscope, Debugger, and DoxyBlot.

14)

```
#include <stdio.h>

int main () {

    int array [10];

    int i;

    printf("Enter 10 integers:\n");

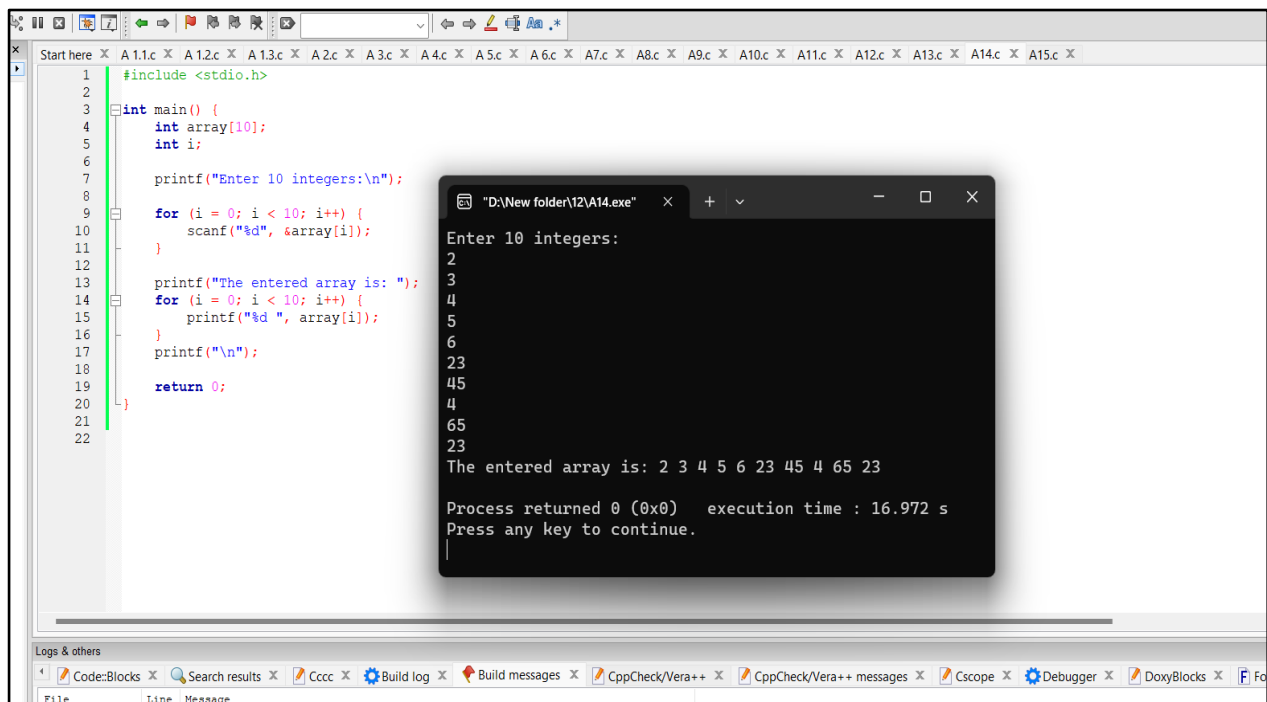
    for (i = 0; i < 10; i++) {
        scanf("%d", &array[i]);
    }

    printf("The entered array is: ");
    for (i = 0; i < 10; i++) {
        printf("%d ", array[i]);
    }

    printf("\n");

    return 0;

}
```



The screenshot shows a C++ IDE with a code editor and a terminal window. The code editor displays the same C++ program as shown in the previous block. The terminal window, titled "D:\New folder\12\A14.exe", shows the program's execution. It prompts the user to "Enter 10 integers:" and then displays the entered values: 2, 3, 4, 5, 6, 23, 45, 4, 65, 23. The output is "The entered array is: 2 3 4 5 6 23 45 4 65 23". The terminal also shows "Process returned 0 (0x0) execution time : 16.972 s" and "Press any key to continue."

15)

```
#include <stdio.h>

int main() {

    int array[10];

    int i, count = 0;

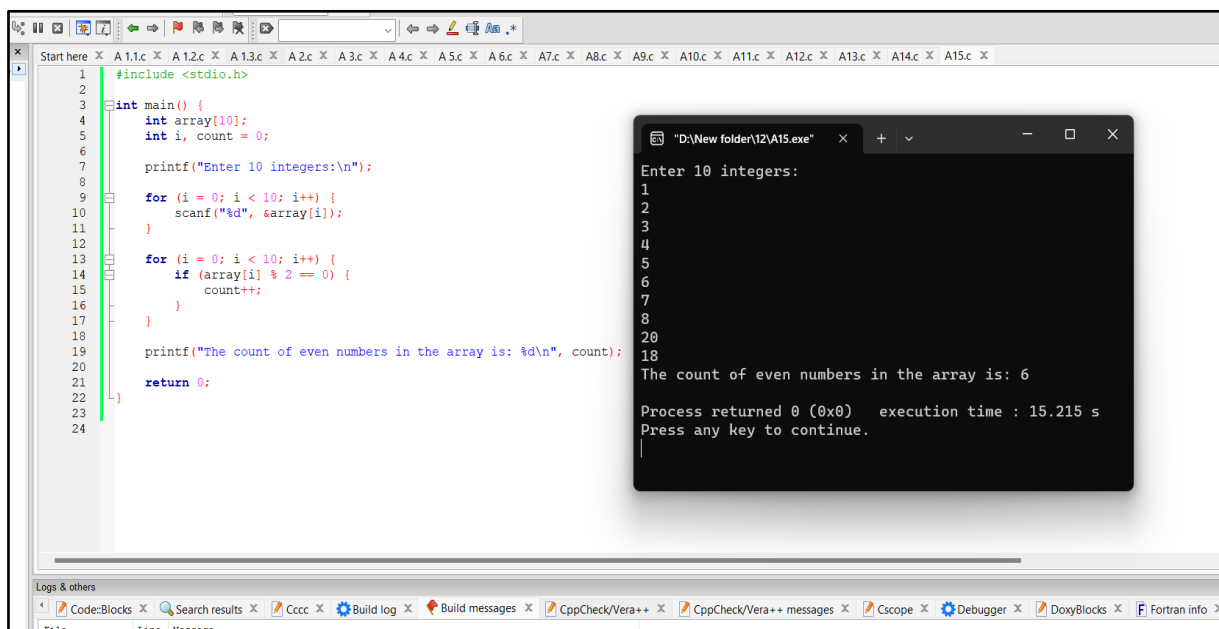
    printf("Enter 10 integers:\n");

    for (i = 0; i < 10; i++) {
        scanf("%d", &array[i]);
    }

    for (i = 0; i < 10; i++) {
        if (array[i] % 2 == 0) {
            count++;
        }
    }

    printf("The count of even numbers in the array is: %d\n", count);

    return 0;
}
```



The screenshot shows a C++ IDE with the code from the previous block. The code is written in a file named A15.c. The code prompts the user to enter 10 integers. The user has entered the following integers: 1, 2, 3, 4, 5, 6, 7, 8, 20, 18. The program then calculates the count of even numbers in the array, which is 6. The output of the program is displayed in a separate window titled "D:\New folder\12\A15.exe". The output shows the entered integers and the count of even numbers. The program returns 0 and the execution time is 15.215 s. The IDE also shows a "Logs & others" panel at the bottom with various tabs like CodeBlocks, Search results, Cccc, Build log, Build messages, CppCheck/Vera++, Cscope, Debugger, DoxyBlocks, and Fortran info.

```
1 #include <stdio.h>
2
3 int main() {
4     int array[10];
5     int i, count = 0;
6
7     printf("Enter 10 integers:\n");
8
9     for (i = 0; i < 10; i++) {
10         scanf("%d", &array[i]);
11     }
12
13     for (i = 0; i < 10; i++) {
14         if (array[i] % 2 == 0) {
15             count++;
16         }
17     }
18
19     printf("The count of even numbers in the array is: %d\n", count);
20
21     return 0;
22 }
23
24
```

Enter 10 integers:

1
2
3
4
5
6
7
8
20
18

The count of even numbers in the array is: 6

Process returned 0 (0x0) execution time : 15.215 s
Press any key to continue.

Section B

1.

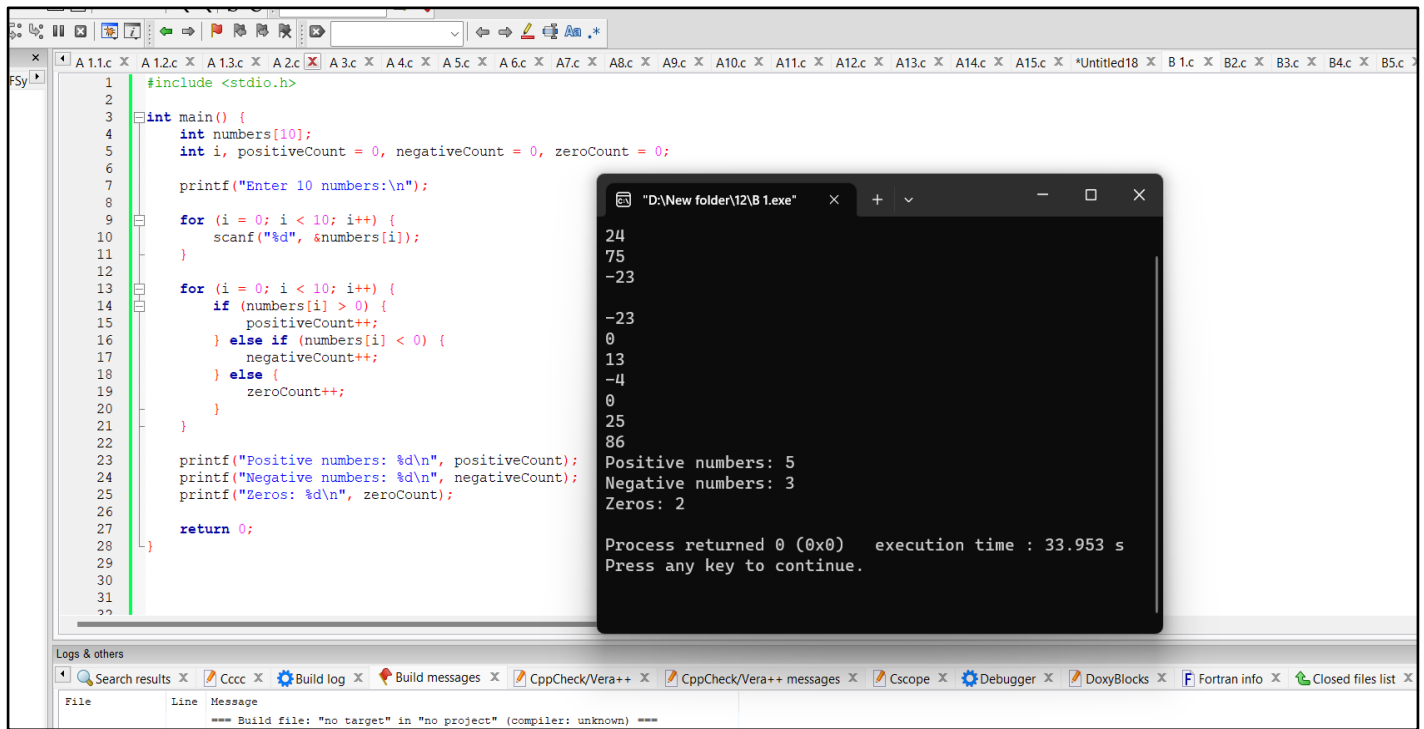
```
#include <stdio.h>

int main() {
    int numbers [10];
    int i, positiveCount = 0, negativeCount = 0, zeroCount = 0;

    printf("Enter 10 numbers:\n");

    for (i = 0; i < 10; i++) {
        scanf("%d", &numbers[i]);
    }
    for (i = 0; i < 10; i++) {
        if (numbers[i] > 0) {
            positiveCount++;
        } else if (numbers[i] < 0) {
            negativeCount++;
        } else {
            zeroCount++;
        }
    }
    printf("Positive numbers: %d\n", positiveCount);
    printf("Negative numbers: %d\n", negativeCount);
    printf("Zeros: %d\n", zeroCount);

    return 0;
}
```



2.

```

#include <stdio.h>

int main() {

    int marks[10];

    int i, totalMarks = 0, maxMarks, minMarks;

    printf("Enter marks of 10 students:\n");

    for (i = 0; i < 10; i++) {
        scanf("%d", &marks[i]);
        totalMarks += marks[i];

        if (i == 0) {
            maxMarks = marks[i];
            minMarks = marks[i];
        } else {
            if (marks[i] > maxMarks) {
                maxMarks = marks[i];
            }
            if (marks[i] < minMarks) {
                minMarks = marks[i];
            }
        }
    }
}

```

```

    }

    if (marks[i] < minMarks) {
        minMarks = marks[i];
    }
}

double averageMarks = (double) totalMarks / 10;

printf("Maximum Marks: %d\n", maxMarks);
printf("Minimum Marks: %d\n", minMarks);
printf("Average Marks: %.2lf\n", averageMarks);

return 0;
}

```

The screenshot shows a C++ IDE with a file named 'A1.1.c' open. The code in the file is as follows:

```

1  #include <stdio.h>
2
3  int main() {
4      int marks[10];
5      int i, totalMarks = 0, maxMarks, minMarks;
6
7      printf("Enter marks of 10 students:\n");
8
9      for (i = 0; i < 10; i++) {
10         scanf("%d", &marks[i]);
11         totalMarks += marks[i];
12
13         if (i == 0) {
14             maxMarks = marks[i];
15             minMarks = marks[i];
16         } else {
17             if (marks[i] > maxMarks) {
18                 maxMarks = marks[i];
19             }
20             if (marks[i] < minMarks) {
21                 minMarks = marks[i];
22             }
23         }
24     }
25
26     double averageMarks = (double) totalMarks / 10;
27
28     printf("Maximum Marks: %d\n", maxMarks);
29     printf("Minimum Marks: %d\n", minMarks);
30     printf("Average Marks: %.2lf\n", averageMarks);
31
32     return 0;
33 }

```

The output window shows the following text:

```

D:\New folder\12\B2.exe
Enter marks of 10 students:
100
24
66
89
75
24
90
67
88
98
Maximum Marks: 100
Minimum Marks: 24
Average Marks: 72.10
Process returned 0 (0x0)   execution time : 22.750 s
Press any key to continue.

```

3.

```
#include <stdio.h>

int main() {

    double prices[10];

    int i, count = 0;

    double total = 0.0;

    printf("Enter prices of 10 items:\n");

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

        scanf("%lf", &prices[i]);

        total += prices[i];

        if (prices[i] > 200) {

            count++;

        }

    }

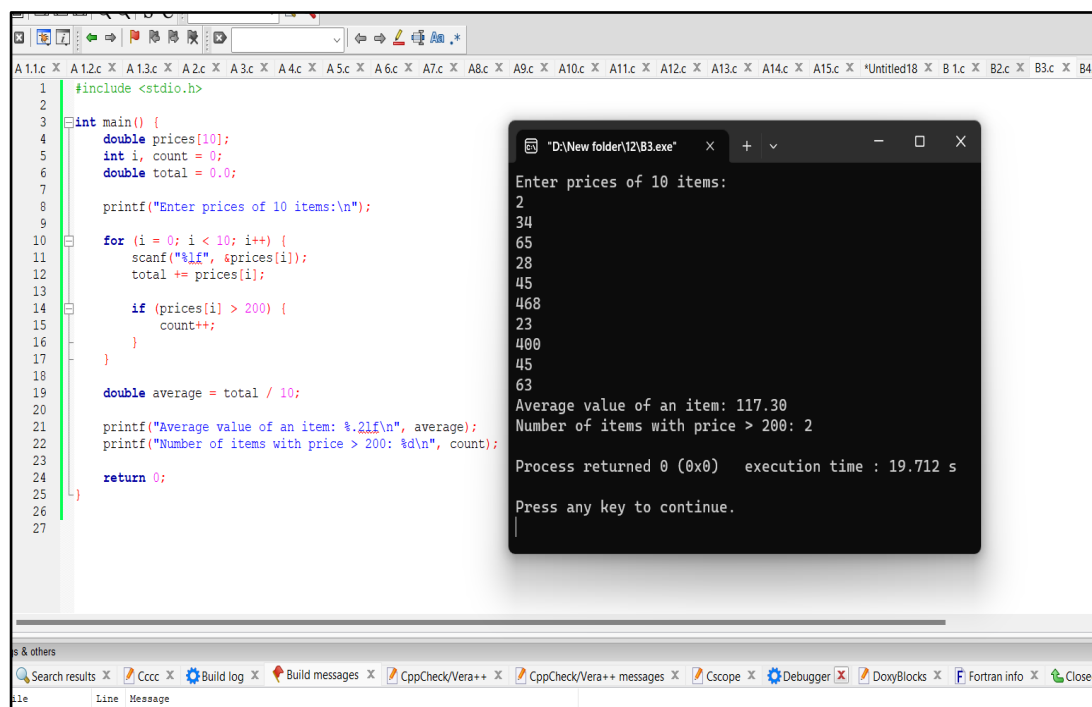
    double average = total / 10;

    printf("Average value of an item: %.2lf\n", average);

    printf("Number of items with price > 200: %d\n", count);

    return 0;

}
```



The screenshot shows a C++ IDE with the code from the previous block. The code is written in a file named 'A11.c'. The code calculates the average of 10 items and counts how many items have a price greater than 200. The output window shows the following results:

```
Enter prices of 10 items:
2
34
65
28
45
468
23
400
45
63
Average value of an item: 117.30
Number of items with price > 200: 2
Process returned 0 (0x0)   execution time : 19.712 s
Press any key to continue.
```

4.

```
#include <stdio.h>

int main() {

    int employeeNo, count = 0;

    float basicSalary;

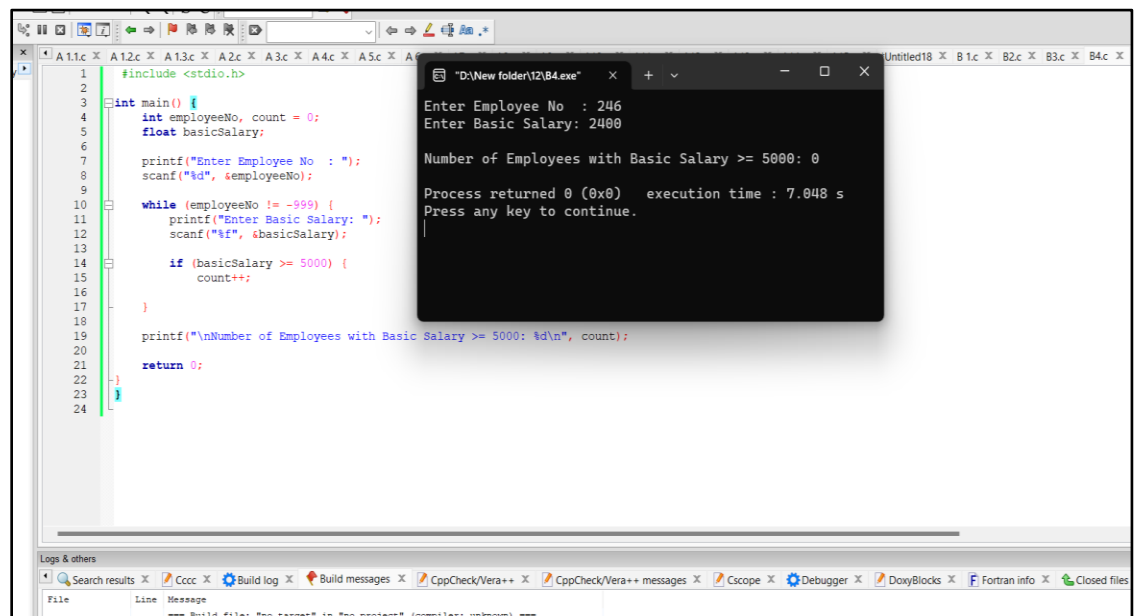
    printf("Enter Employee No : ");
    scanf("%d", &employeeNo);

    while (employeeNo != -999) {
        printf("Enter Basic Salary: ");
        scanf("%f", &basicSalary);

        if (basicSalary >= 5000) {
            count++;
        }
    }

    printf("\nNumber of Employees with Basic Salary >= 5000: %d\n", count);

    return 0;
}
```



The screenshot displays a C++ IDE with a code editor on the left and a console window on the right. The code in the editor is the same as shown in the previous block. The console window shows the following output:

```
Enter Employee No : 246
Enter Basic Salary: 2400

Number of Employees with Basic Salary >= 5000: 0

Process returned 0 (0x0)   execution time : 7.048 s
Press any key to continue.
```

5.

```
#include <stdio.h>

int main () {

    int employeeNo, count = 0, overtimeCount = 0;

    double hoursWorked, overtimePayment, totalOvertimePayment = 0.0;

    printf("Enter employee number and hours worked :\n");

    scanf("%d", &employeeNo);

    while (employeeNo != -999) {
        scanf("%lf", &hoursWorked);

        if (hoursWorked > 40) {
            overtimePayment = 150 * 40 + 200 * (hoursWorked - 40);
        } else {
            overtimePayment = 150 * hoursWorked;
        }

        printf("Employee number: %d\n", employeeNo);
        printf("Overtime payment: %.2lf\n", overtimePayment);

        totalOvertimePayment += overtimePayment;
        count++;
        if (overtimePayment > 4000) {
            overtimeCount++;
        }
        scanf("%d", &employeeNo);
    }

    double percentageExceeding4000 = (double) overtimeCount / count * 100;
```



```

printf("\nSummary:\n");

printf("Total employees: %d\n", count);

printf("Total overtime payment: %.2lf\n", totalOvertimePayment);

printf("Percentage of employees with overtime payment exceeding Rs. 4000:
%.2lf%%\n", percentageExceeding4000);

return 0;
}

```

The screenshot shows a C++ IDE with a source file and a terminal window. The source file contains the following code:

```

1  #include <stdio.h>
2
3  int main() {
4      int employeeNo, count = 0, overtimeCount = 0;
5      double hoursWorked, overtimePayment, totalOvertimePayment = 0.0;
6
7      printf("Enter employee number and hours worked :\n");
8
9      scanf("%d", &employeeNo);
10
11     while (employeeNo != -999) {
12         scanf("%lf", &hoursWorked);
13
14         if (hoursWorked > 40) {
15             overtimePayment = 150 * 40 + 200 * (hoursWorked - 40);
16         } else {
17             overtimePayment = 150 * hoursWorked;
18         }
19         printf("Employee number: %d\n", employeeNo);
20         printf("Overtime payment: %.2lf\n", overtimePayment);
21
22         totalOvertimePayment += overtimePayment;
23         count++;
24         if (overtimePayment > 4000) {
25             overtimeCount++;
26         }
27         scanf("%d", &employeeNo);
28     }
29     double percentageExceeding4000 = (double) overtimeCount / count * 100;
30
31     printf("\nSummary:\n");
32     printf("Total employees: %d\n", count);

```

The terminal window shows the output of the program:

```

D:\New folder\12\B5. x + v - □ ×
Enter employee number and hours worked :
245
12000
Employee number: 245
Overtime payment: 2398000.00

```

The IDE interface includes a toolbar at the top, a tab bar with multiple open files, and a bottom panel with tabs for 'Logs & others', 'Search results', 'Cccc', 'Build log', 'Build messages', 'CppCheck/Vera++', 'CppCheck/Vera++ messages', 'Cscope', 'Debugger', 'DoxyBlocks', 'Fortran info', and 'Closed f'.