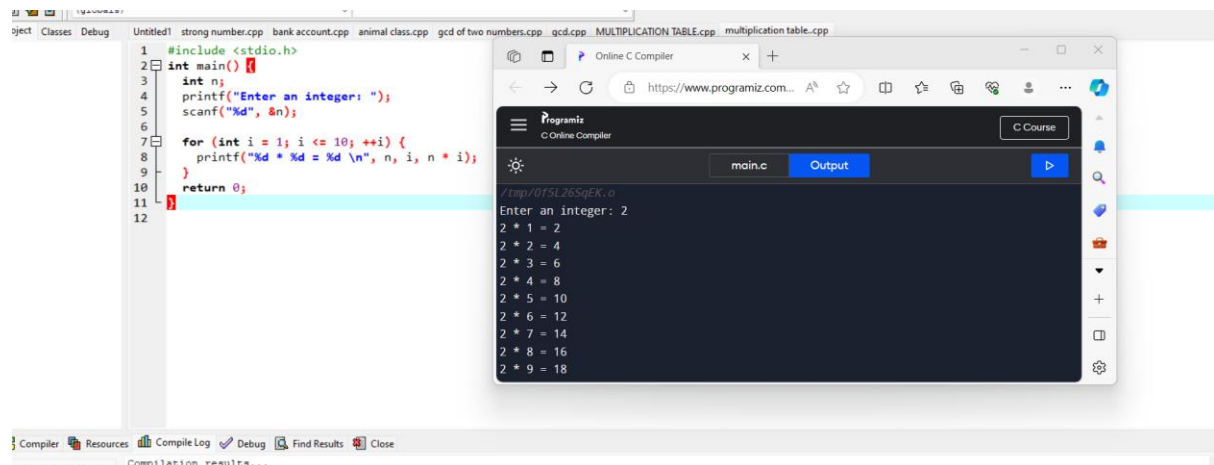


MODEL PRACTICAL EXAM

1)MULTIPLICATION TABLE



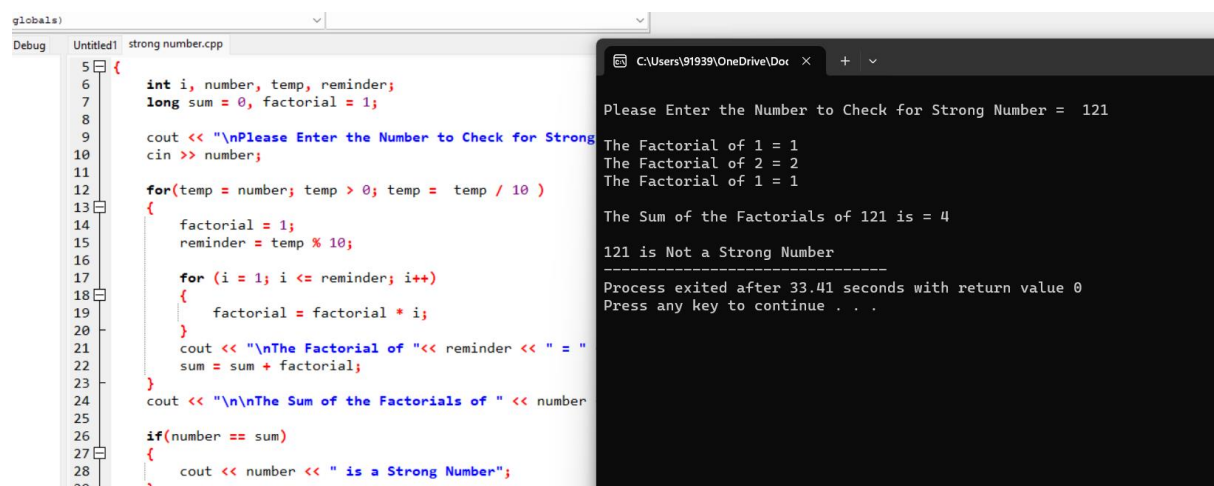
The screenshot shows a C++ IDE with a file named 'multiplication table.cpp'. The code is as follows:

```
1 #include <stdio.h>
2 int main()
3 {
4     int n;
5     printf("Enter an integer: ");
6     scanf("%d", &n);
7     for (int i = 1; i <= 10; ++i) {
8         printf("%d * %d = %d \n", n, i, n * i);
9     }
10 }
11 return 0;
12
```

The output of the program is displayed in a terminal window:

```
Enter an integer: 2
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
```

2)STRONG NUMBER



The screenshot shows a C++ IDE with a file named 'strong number.cpp'. The code is as follows:

```
5 {
6     int i, number, temp, remainder;
7     long sum = 0, factorial = 1;
8
9     cout << "\nPlease Enter the Number to Check for Strong Number";
10    cin >> number;
11
12    for(temp = number; temp > 0; temp = temp / 10)
13    {
14        factorial = 1;
15        remainder = temp % 10;
16
17        for (i = 1; i <= remainder; i++)
18        {
19            factorial = factorial * i;
20        }
21        cout << "\nThe Factorial of " << remainder << " = " << factorial << "\n";
22        sum = sum + factorial;
23    }
24    cout << "\n\nThe Sum of the Factorials of " << number << " = " << sum << "\n";
25
26    if(number == sum)
27    {
28        cout << number << " is a Strong Number";
29    }
30 }
```

The output of the program is displayed in a terminal window:

```
Please Enter the Number to Check for Strong Number = 121
The Factorial of 1 = 1
The Factorial of 2 = 2
The Factorial of 1 = 1

The Sum of the Factorials of 121 is = 4

121 is Not a Strong Number
-----
Process exited after 33.41 seconds with return value 0
Press any key to continue . . .
```

3)PATTERN

```
8      std::cin >> n;
9
10     // Loop to iterate through rows
11     for (int i = 1; i <= n; ++i) {
12         // Loop to print spaces
13         for (int j = 1; j <= n - i; ++j) {
14             std::cout << " ";
15         }
16
17         // Loop to print increasing numbers
18         for (int k = 1; k <= i; ++k) {
19             std::cout << k;
20         }
21
22         // loop to print decreasing numbers
23         for (int l = i - 1; l >= 1; --l) {
24             std::cout << l;
25         }
26
27         // Move to the next line after each row
28         std::cout << "\n";
29     }
30
31     return 0;
32 }
33
```

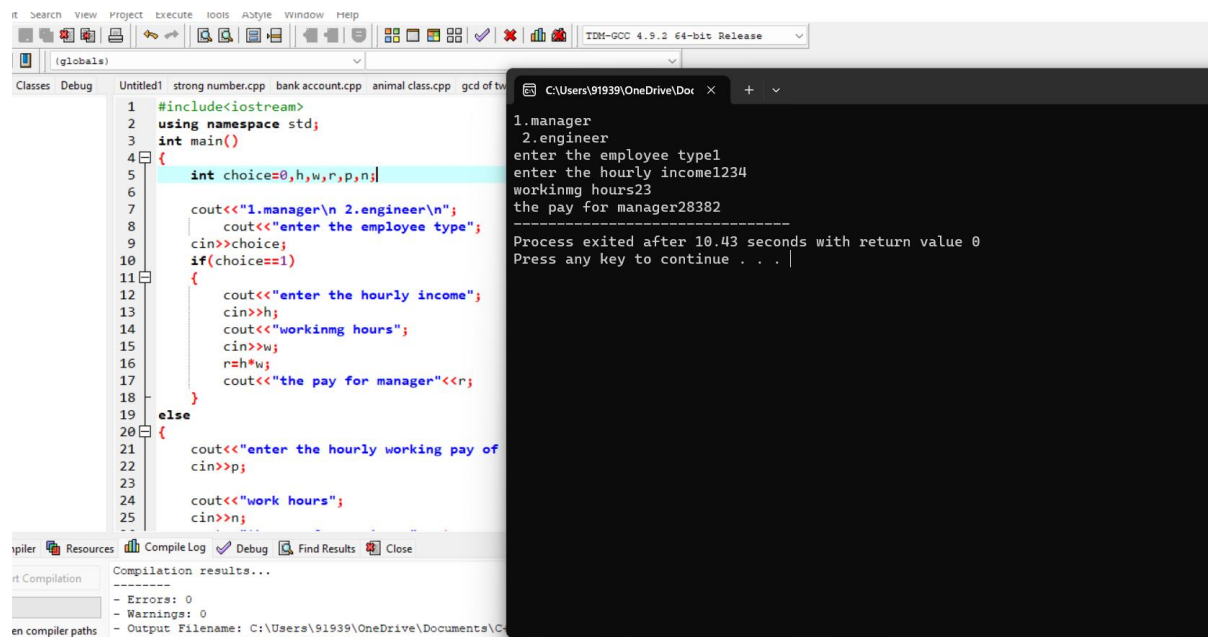
```
C:\mp\kpcwkr97wy.o
Enter the number of rows: 5
1
121
12321
1234321
123454321
```

4)GCD

```
1  #include <stdio.h>
2  int main()
3  {
4      int n1, n2, i, gcd;
5
6      printf("Enter two integers: ");
7      scanf("%d %d", &n1, &n2);
8
9      for(i=1; i <= n1 && i <= n2; ++i)
10     {
11         if(n1%i==0 && n2%i==0)
12             gcd = i;
13     }
14
15     printf("G.C.D of %d and %d is %d", n1, n2, gcd);
16
17     return 0;
18 }
19
```

```
C:\Users\91939\OneDrive\Doc
Enter two integers: 12
23
G.C.D of 12 and 23 is 1
-----
Process exited after 26.97 seconds with return value 0
Press any key to continue . . .
```

5) CLACULATE PAY OF EMPLOYEE

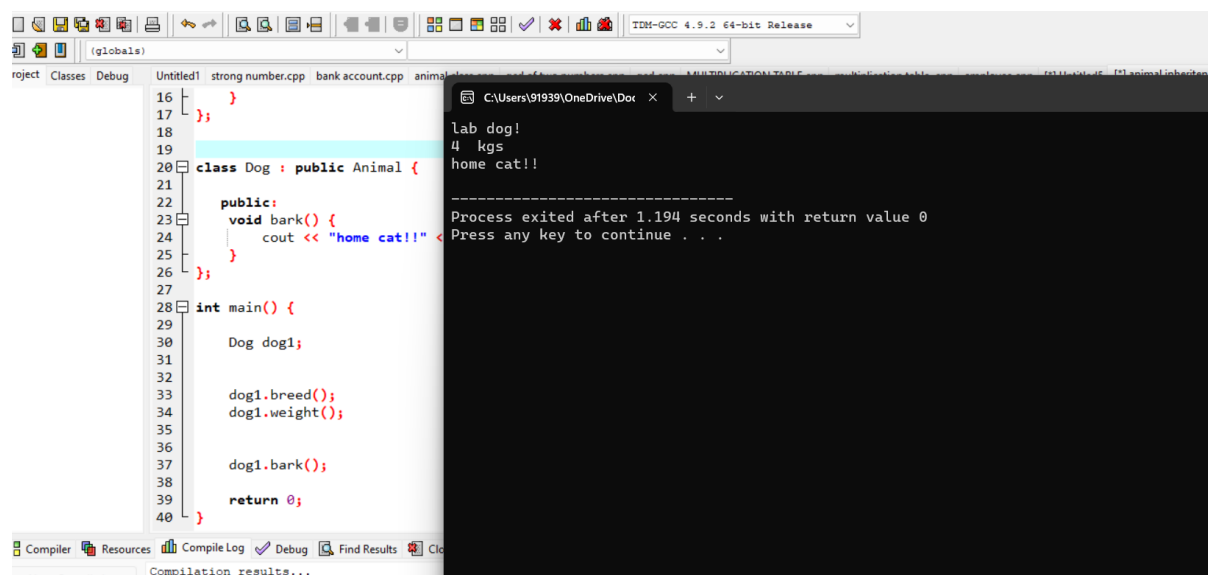


```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int choice=0,h,w,r,p,n;
6
7     cout<<"1.manager\n 2.engineer\n";
8     cout<<"enter the employee type";
9     cin>>choice;
10    if(choice==1)
11    {
12        cout<<"enter the hourly income";
13        cin>>h;
14        cout<<"workinmg hours";
15        cin>>w;
16        r=h*w;
17        cout<<"the pay for manager"<<r;
18    }
19    else
20    {
21        cout<<"enter the hourly working pay of";
22        cin>>p;
23
24        cout<<"work hours";
25        cin>>n;
26    }
27 }
```

1.manager
2.engineer
enter the employee type1
enter the hourly income1234
workinmg hours23
the pay for manager28382

Process exited after 10.43 seconds with return value 0
Press any key to continue . . . |

6)ANIMAL CLASS

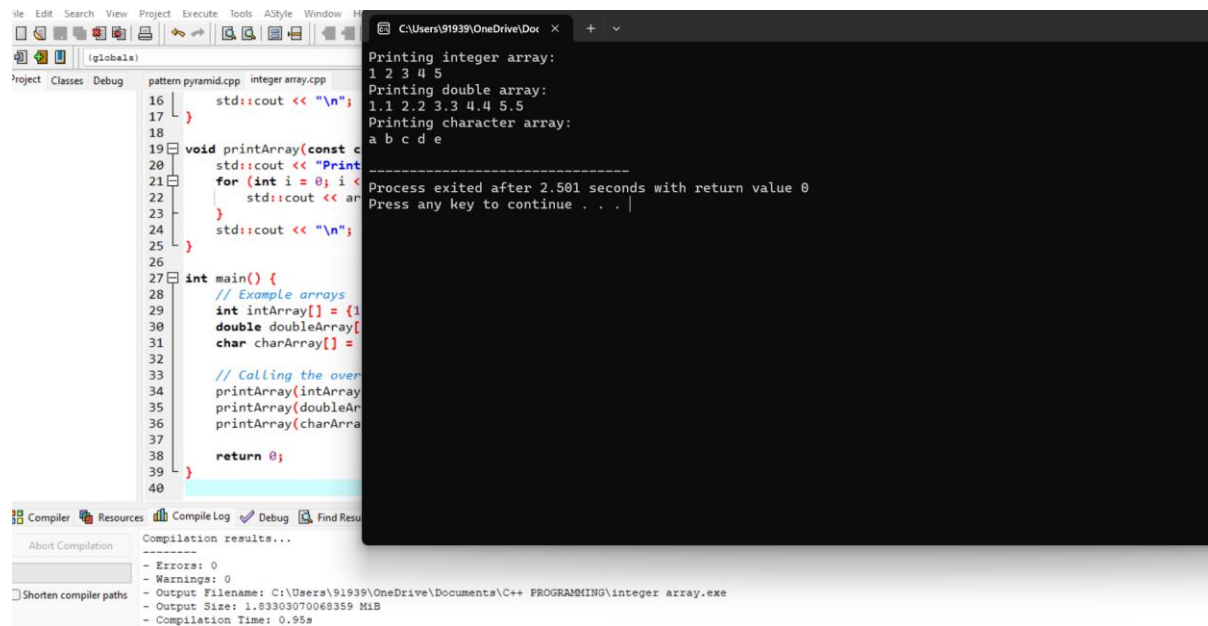


```
16 }
17 };
18
19
20 class Dog : public Animal {
21
22 public:
23     void bark() {
24         cout << "home cat!!" << endl;
25     }
26 };
27
28 int main() {
29
30     Dog dog1;
31
32     dog1.breed();
33     dog1.weight();
34
35     dog1.bark();
36
37     return 0;
38 }
39
40 }
```

lab dog!
4 kgs
home cat!!

Process exited after 1.194 seconds with return value 0
Press any key to continue . . .

7) INTEGER ARRAY ,DOUBLE ARRAY ,CHARACTER ARRAY



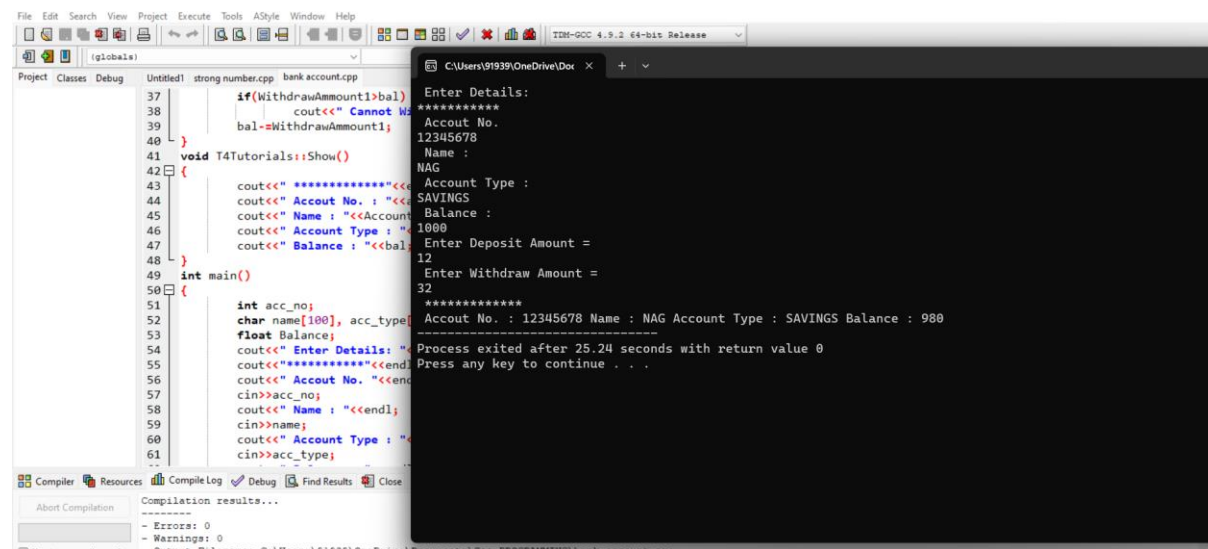
The screenshot shows a C++ IDE with a file named `integer array.cpp`. The code defines a `printArray` function that takes a constant array and its size, and prints its elements. The `main` function demonstrates this with an integer array `{1, 2, 3, 4, 5}`, a double array `{1.1, 2.2, 3.3, 4.4, 5.5}`, and a character array `"a b c d e"`.

```
16 |         std::cout << "\n";
17 |     }
18 |
19 |     void printArray(const T arr, int size) {
20 |         std::cout << "Print Array: ";
21 |         for (int i = 0; i < size; i++) {
22 |             std::cout << arr[i] << " ";
23 |         }
24 |         std::cout << "\n";
25 |     }
26 |
27 |     int main() {
28 |         // Example arrays
29 |         int intArray[] = {1, 2, 3, 4, 5};
30 |         double doubleArray[] = {1.1, 2.2, 3.3, 4.4, 5.5};
31 |         char charArray[] = "a b c d e";
32 |
33 |         // Calling the overloads
34 |         printArray(intArray, 5);
35 |         printArray(doubleArray, 5);
36 |         printArray(charArray, 5);
37 |
38 |         return 0;
39 |     }
40 | }
```

The output window shows the following text:

```
Printing integer array:
1 2 3 4 5
Printing double array:
1.1 2.2 3.3 4.4 5.5
Printing character array:
a b c d e
-----
Process exited after 2.501 seconds with return value 0
Press any key to continue . . .
```

8) BANK ACCOUNT



The screenshot shows a C++ IDE with a file named `bank account.cpp`. The code defines a `T4Tutorials` class with a `Show` method that displays account details. The `main` function creates an account and demonstrates the `Show` method.

```
37 |         if(WithdrawAmount > bal)
38 |             cout << " Cannot Withdraw\n";
39 |         bal = WithdrawAmount;
40 |     }
41 |     void T4Tutorials::Show()
42 |     {
43 |         cout << "*****\n";
44 |         cout << "Account No. : " << acc_no << "\n";
45 |         cout << "Name : " << name << "\n";
46 |         cout << "Account Type : " << acc_type << "\n";
47 |         cout << "Balance : " << bal << "\n";
48 |     }
49 |
50 |     int main()
51 |     {
52 |         int acc_no;
53 |         char name[100], acc_type[50];
54 |         float Balance;
55 |         cout << "Enter Details: ";
56 |         cout << "*****\n";
57 |         cout << "Account No. : ";
58 |         cin >> acc_no;
59 |         cout << "Name : ";
60 |         cin >> name;
61 |         cout << "Account Type : ";
62 |         cin >> acc_type;
```

The output window shows the following text:

```
Enter Details:
*****
Account No. :
12345678
Name :
NAG
Account Type :
SAVINGS
Balance :
1000
Enter Deposit Amount =
12
Enter Withdraw Amount =
32
*****
Account No. : 12345678 Name : NAG Account Type : SAVINGS Balance : 980
-----
Process exited after 25.24 seconds with return value 0
Press any key to continue . . .
```