Programiz
C Online Compiler

Programiz PRO

main.c   Output   ☾ ⤴ ▷

```c
#include <stdio.h>

int main() {
    int num = 10;
    int *ptr = &num; // Pointer
        storing the address of
        num

    printf("Value of num: %d\n",
        num);
    printf("Address of num: %p\n"
        , &num);
    printf("Value stored in
        pointer ptr: %p\n", ptr);
    printf("Value pointed to by
        ptr: %d\n", *ptr);
printf("bindhu t j\n") ;
    return 0;
}
```

Run

Programiz
C Online Compiler

Programiz PRO

main.c | Output

Value of num: 10

Address of num: 0x7ffe321aedf4

Value stored in pointer ptr:

   0x7ffe321aedf4

Value pointed to by ptr: 10

bindhu t j

=== Code Execution Successful ===

main.c | Output

```c
#include <stdio.h>

int main() {
    int arr[] = {10, 20, 30, 40};
    int *ptr = arr;  // Pointer to
         the first element of the
         array

    for (int i = 0; i < 4; i++) {
        printf("Value at arr[%d]:
            %d\n", i, *(ptr + i));
        printf ("bindhu t j");
    }
return 0;
}
```

Run

main.c | **Output**

```
Value at arr[0]: 10
bindhu t jValue at arr[1]: 20
bindhu t jValue at arr[2]: 30
bindhu t jValue at arr[3]: 40
bindhu t j

=== Code Execution Successful ===
```

Programiz
C Online Compiler

Programiz PRO

main.c	Output

🌙 ⤴ ▷

```c
1   #include<stdio.h>
2 ▾ int main() {
3       int num = 100;
4       int *ptr = &num;      //
            Pointer to num
5       int **ptr2 = &ptr;      //
            Pointer to pointer
6
7       printf("Value of num: %d\n",
            num);
8       printf("Value using ptr: %d\n",
            *ptr);
9       printf("Value using ptr2: %d\n"
            , **ptr2);
10  printf ("bindhu t j");
11      return 0;
12  }
```

Run

Programiz
C Online Compiler

Programiz PRO

| main.c | Output |

```
Value of num: 100
Value using ptr: 100
Value using ptr2: 100
bindhu t j


=== Code Execution Successful ===
```

Programiz
C Online Compiler

Programiz PRO

main.c | Output

```c
#include <stdio.h>

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

int main() {
    int value = 5;

    printf("Before increment: %d\n", value);
    increment(&value);  // Pass address of value to the function
    printf("After increment: %d\n", value);
    printf ("bindhu t j");
    return 0;
}
```

Run

Programiz
C Online Compiler

Programiz PRO

main.c | Output

```
Before increment: 5
After increment: 6
bindhu t j


=== Code Execution Successful ===
```

Programiz
C Online Compiler

Programiz PRO

main.c   Output   ▷

```c
1   #include <stdio.h>
2   int main() {
3       int numbers[5] = {10, 20, 30,
            40, 50};
4
5       for (int i = 0; i < 5; i++) {
6           printf("Element at index %d
                : %d\n", i, numbers[i]
                );
7       }
8       printf ("bindhu t j");
9       return 0;
10  }
```

Run

Programiz
C Online Compiler

Programiz PRO

main.c	Output

```
Element at index 0: 10
Element at index 1: 20
Element at index 2: 30
Element at index 3: 40
Element at index 4: 50
bindhu t j


=== Code Execution Successful ===
```

Programiz
C Online Compiler

Programiz PRO

main.c   Output

```c
1   // Online C compiler to run C
        program online
2   #include <stdio.h>
3   int main() {
4       printf("bindhu t j");
5   int num =123, sum=0;
6   while(num!=0) {
7       num/=10;
8   }
9   printf ("sum of digits =%d\n",sum);
10      return 0;
11  }
```

Run

Programiz
C Online Compiler

Programiz PRO

main.c   Output

```c
1   // Online C compiler to run C
        program online
2   #include <stdio.h>
3 ▾ int add(int a, int b){
4   return a+b;
5   }
6 ▾ int main() {
7       int result=add(5, 10);
8       printf("sum=%d\n,result");
9       return 0;
10  }
```

Run

Programiz
C Online Compiler

Programiz PRO

main.c | Output

```
a=10,b=5


=== Code Execution Successful ===
```

Programiz
C Online Compiler

Programiz PRO

main.c | Output

```
bindhu t jsum of digits =0



=== Code Execution Successful ===
```