C Programming Practical Programs

100 Practical Programs for Beginners, Students & Coding Practice

This PDF contains the C programs supplied in the uploaded C Programs document, organized by category with program code and sample output.

C Programs

Introduction

C Programs are the best way to understand programming concepts through practical implementation.

This section contains a collection of commonly used C programs for beginners, students, and interview preparation.

The programs are organized into different categories to help learners practice concepts step by step and improve their coding skills.

Program Categories

This collection contains programs from the following categories:

• Basic Programs

• Number Programs

• Pattern Programs

• Array Programs

• String Programs

• Function Programs

• Pointer Programs

• File Handling Programs

• Interview Programs

Basic C Programs

1. Hello World Program

Program

\#include \<stdio.h>

```
int main()
{
printf("Hello, World!");

return 0;
}
```

Output

Hello, World!

2. Print an Integer

Program

\#include \<stdio.h>

```
int main()
{
int number = 25;

printf("%d", number);
```

```
return 0;
}
```

Output

25

3. Add Two Numbers

Program

\#include \<stdio.h>

```
int main()
{
int a = 10, b = 20;

printf("Sum = %d", a + b);

return 0;
}
```

Output

Sum = 30

4. Swap Two Numbers

Program

\#include \<stdio.h>

```
int main()
{
int a = 10, b = 20, temp;

temp = a;
a = b;
b = temp;

printf("a = %d\n", a);
printf("b = %d", b);

return 0;
}
```

Output

```
a = 20
b = 10
```

5. Check Even or Odd Number

Program

\#include \<stdio.h>

```
int main()
{
int num = 8;

if(num % 2 == 0)
printf("Even");
else
printf("Odd");

return 0;
```

```
}
```

Output

Even

6. Find the Largest of Two Numbers

Program

\#include \<stdio.h>

```
int main()
{
int a = 25, b = 15;

if(a > b)
printf("%d is Largest", a);
else
printf("%d is Largest", b);

return 0;
}
```

Output

25 is Largest

7. Find the Largest of Three Numbers

Program

\#include \<stdio.h>

```
int main()
{
int a = 20, b = 35, c = 15;

if(a >= b && a >= c)
printf("%d is Largest", a);
else if(b >= a && b >= c)
printf("%d is Largest", b);
else
printf("%d is Largest", c);

return 0;
}
```

Output

35 is Largest

8. Find the Smallest of Three Numbers

Program

\#include \<stdio.h>

```
int main()
{
int a = 20, b = 35, c = 15;

if(a <= b && a <= c)
printf("%d is Smallest", a);
else if(b <= a && b <= c)
printf("%d is Smallest", b);
```

```
else
printf("%d is Smallest", c);

return 0;
}
```

Output

15 is Smallest

9. Check Leap Year

Program

\#include \<stdio.h>

```
int main()
{
int year = 2024;

if((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
printf("Leap Year");
else
printf("Not a Leap Year");

return 0;
}
```

Output

Leap Year

10. Check Positive, Negative, or Zero

Program

\#include \<stdio.h>

```
int main()
{
int num = -8;

if(num > 0)
printf("Positive");
else if(num < 0)
printf("Negative");
else
printf("Zero");

return 0;
}
```

Output

Negative

Number Programs

11. Find the Factorial of a Number

Program

\#include \<stdio.h>

```
int main()
```

```
{
int i, n = 5, fact = 1;

for(i = 1; i <= n; i++)
fact *= i;

printf("Factorial = %d", fact);

return 0;
}
```

Output

Factorial = 120

12. Generate Fibonacci Series

Program

\#include \<stdio.h>

```
int main()
{
int a = 0, b = 1, c, i;

printf("%d %d ", a, b);

for(i = 1; i <= 8; i++)
{
c = a + b;
printf("%d ", c);
a = b;
b = c;
}

return 0;
}
```

Output

0 1 1 2 3 5 8 13 21 34

13. Check Prime Number

Program

\#include \<stdio.h>

```
int main()
{
int n = 13, i, prime = 1;

for(i = 2; i < n; i++)
{
if(n % i == 0)
{
prime = 0;
break;
}
}

if(prime)
printf("Prime Number");
else
printf("Not a Prime Number");

return 0;
```

```
}
```

Output

Prime Number

14. Check Palindrome Number

Program

\#include \<stdio.h>

```
int main()
{
int n = 121, temp, rev = 0, rem;

temp = n;

while(n != 0)
{
rem = n % 10;
rev = rev * 10 + rem;
n /= 10;
}

if(temp == rev)
printf("Palindrome");
else
printf("Not Palindrome");

return 0;
}
```

Output

Palindrome

15. Check Armstrong Number

Program

\#include \<stdio.h>

```
int main()
{
int n = 153, temp, rem, sum = 0;

temp = n;

while(n != 0)
{
rem = n % 10;
sum += rem * rem * rem;
n /= 10;
}

if(temp == sum)
printf("Armstrong Number");
else
printf("Not an Armstrong Number");

return 0;
}
```

Output

Armstrong Number

16. Reverse a Number

Program

\#include \<stdio.h>

```
int main()
{
int n = 12345, rev = 0, rem;

while(n != 0)
{
rem = n % 10;
rev = rev * 10 + rem;
n /= 10;
}

printf("Reverse = %d", rev);

return 0;
}
```

Output

Reverse = 54321

17. Sum of Digits

Program

\#include \<stdio.h>

```
int main()
{
int n = 1234, sum = 0;

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

printf("Sum = %d", sum);

return 0;
}
```

Output

Sum = 10

18. Count the Number of Digits

Program

\#include \<stdio.h>

```
int main()
{
int n = 12345, count = 0;

while(n != 0)
{
count++;
```

```
n /= 10;
}

printf("Digits = %d", count);

return 0;
}
```

Output

Digits = 5

19. Check Perfect Number

Program

\#include \<stdio.h>

```
int main()
{
int n = 28, i, sum = 0;

for(i = 1; i < n; i++)
{
if(n % i == 0)
sum += i;
}

if(sum == n)
printf("Perfect Number");
else
printf("Not a Perfect Number");

return 0;
}
```

Output

Perfect Number

20. Check Strong Number

Program

\#include \<stdio.h>

```
int main()
{
int n = 145, temp, rem, sum = 0, i, fact;

temp = n;

while(n != 0)
{
rem = n % 10;
fact = 1;

for(i = 1; i <= rem; i++)
fact *= i;

sum += fact;
n /= 10;
}

if(sum == temp)
printf("Strong Number");
```

```
else
printf("Not a Strong Number");

return 0;
}
```

Output

Strong Number

Pattern Programs

21. Right Triangle Star Pattern

Program

\#include \<stdio.h>

```
int main()
{
int i, j;

for(i = 1; i <= 5; i++)
{
for(j = 1; j <= i; j++)
printf("* ");

printf("\n");
}

return 0;
}
```

Output

*

* *

* * *

* * * *

* * * * *

22. Inverted Right Triangle Pattern

Program

\#include \<stdio.h>

```
int main()
{
int i, j;

for(i = 5; i >= 1; i-)
{
for(j = 1; j <= i; j++)
printf("* ");

printf("\n");
}

return 0;
}
```

Output

* * * * *

* * * *

* * *

* *

*

23. Square Star Pattern

Program

\#include \<stdio.h>

```
int main()
{
int i, j;

for(i = 1; i <= 5; i++)
{
for(j = 1; j <= 5; j++)
printf("* ");

printf("\n");
}

return 0;
}
```

Output

* * * * *

* * * * *

* * * * *

* * * * *

* * * * *

24. Number Triangle Pattern

Program

\#include \<stdio.h>

```
int main()
{
int i, j;

for(i = 1; i <= 5; i++)
{
for(j = 1; j <= i; j++)
printf("%d ", j);

printf("\n");
}

return 0;
}
```

Output

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

25. Floyd's Triangle

Program

\#include \<stdio.h>

```
int main()
{
int i, j, num = 1;

for(i = 1; i <= 5; i++)
{
for(j = 1; j <= i; j++)
printf("%d ", num++);

printf("\n");
}

return 0;
}
```

Output

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

26. Inverted Number Triangle

Program

\#include \<stdio.h>

```
int main()
{
int i, j;

for(i = 5; i >= 1; i-)
{
for(j = 1; j <= i; j++)
printf("%d ", j);

printf("\n");
}

return 0;
}
```

Output

1 2 3 4 5

1 2 3 4

1 2 3

1 2

1

27. Pyramid Star Pattern

Program

\#include \<stdio.h>

```
int main()
{
int i, j, space;

for(i = 5; i >= 1; i-)
{
for(space = 1; space <= 5 - i; space++)
printf(" ");

for(j = 1; j <= (2 * i - 1); j++)
printf("*");

printf("\n");
}

return 0;
}
```

Output

****

****

*****

***

*

28. Inverted Pyramid Star Pattern

Program

\#include \<stdio.h>

```
int main()
{
int i, j, space;

for(i = 5; i >= 1; i-)
{
for(space = 1; space <= 5 - i; space++)
printf(" ");

for(j = 1; j <= (2 * i - 1); j++)
printf("*");

printf("\n");
```

```
}

return 0;
}
```

Output

****

****

******

***

*

29. Hollow Square Pattern

Program

\#include \<stdio.h>

```
int main()
{
int i, j;

for(i = 1; i <= 5; i++)
{
for(j = 1; j <= 5; j++)
{
if(i == 1 || i == 5 || j == 1 || j == 5)
printf("* ");
else
printf(" ");
}

printf("\n");
}

return 0;
}
```

Output

* * * * *

* *

* *

* *

* * * * *

30. Diamond Star Pattern

Program

\#include \<stdio.h>

```
int main()
{
int i, j, space;

for(i = 1; i <= 5; i++)
{
```

```
for(space = 1; space <= 5 - i; space++)
printf(" ");

for(j = 1; j <= (2 * i - 1); j++)
printf("*");

printf("\n");
}

for(i = 4; i >= 1; i-)
{
for(space = 1; space <= 5 - i; space++)
printf(" ");

for(j = 1; j <= (2 * i - 1); j++)
printf("*");

printf("\n");
}

return 0;
}
```

Output

*

***

*****

****

****

****

*****

***

*

Array Programs

31. Print Array Elements

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {10, 20, 30, 40, 50};
int i;

for(i = 0; i < 5; i++)
printf("%d ", arr[i]);

return 0;
}
```

Output

10 20 30 40 50

32. Sum of Array Elements

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {10, 20, 30, 40, 50};
int i, sum = 0;

for(i = 0; i < 5; i++)
sum += arr[i];

printf("Sum = %d", sum);

return 0;
}
```

Output

Sum = 150

33. Find Largest Element in an Array

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {12, 45, 7, 89, 23};
int i, largest = arr[0];

for(i = 1; i < 5; i++)
{
if(arr[i] > largest)
largest = arr[i];
}

printf("Largest = %d", largest);

return 0;
}
```

Output

Largest = 89

34. Find Smallest Element in an Array

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {12, 45, 7, 89, 23};
int i, smallest = arr[0];

for(i = 1; i < 5; i++)
{
if(arr[i] < smallest)
smallest = arr[i];
}
```

```
printf("Smallest = %d", smallest);

return 0;
}
```

Output

Smallest = 7

35. Calculate Average of Array Elements

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {10, 20, 30, 40, 50};
int i, sum = 0;
float average;

for(i = 0; i < 5; i++)
sum += arr[i];

average = (float)sum / 5;

printf("Average = %.2f", average);

return 0;
}
```

Output

Average = 30.00

36. Reverse an Array

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {10, 20, 30, 40, 50};
int i;

printf("Reversed Array: ");

for(i = 4; i >= 0; i-)
printf("%d ", arr[i]);

return 0;
}
```

Output

Reversed Array: 50 40 30 20 10

37. Sort Array in Ascending Order

Program

\#include \<stdio.h>

```
int main()
{
```

```
int arr[] = {45, 12, 89, 7, 23};
int i, j, temp;

for(i = 0; i < 5; i++)
{
for(j = i + 1; j < 5; j++)
{
if(arr[i] > arr[j])
{
temp = arr[i];
arr[i] = arr[j];
arr[j] = temp;
}
}
}

for(i = 0; i < 5; i++)
printf("%d ", arr[i]);

return 0;
}
```

Output

7 12 23 45 89

38. Sort Array in Descending Order

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {45, 12, 89, 7, 23};
int i, j, temp;

for(i = 0; i < 5; i++)
{
for(j = i + 1; j < 5; j++)
{
if(arr[i] < arr[j])
{
temp = arr[i];
arr[i] = arr[j];
arr[j] = temp;
}
}
}

for(i = 0; i < 5; i++)
printf("%d ", arr[i]);

return 0;
}
```

Output

89 45 23 12 7

39. Search an Element in an Array

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {10, 20, 30, 40, 50};
int i, key = 30, found = 0;

for(i = 0; i < 5; i++)
{
if(arr[i] == key)
{
found = 1;
break;
}
}

if(found)
printf("Element Found");
else
printf("Element Not Found");

return 0;
}
```

Output

Element Found

40. Copy One Array to Another

Program

\#include \<stdio.h>

```
int main()
{
int arr1[] = {10, 20, 30, 40, 50};
int arr2[5];
int i;

for(i = 0; i < 5; i++)
arr2[i] = arr1[i];

for(i = 0; i < 5; i++)
printf("%d ", arr2[i]);

return 0;
}
```

Output

10 20 30 40 50

41. Find the Second Largest Element in an Array

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {12, 45, 7, 89, 23};
int i, largest, second;

largest = second = arr[0];

for(i = 1; i < 5; i++)
{
```

```
if(arr[i] > largest)
{
second = largest;
largest = arr[i];
}
else if(arr[i] > second && arr[i] != largest)
{
second = arr[i];
}
}

printf("Second Largest = %d", second);

return 0;
}
```

Output

Second Largest = 45

42. Find the Second Smallest Element in an Array

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {12, 45, 7, 89, 23};
int i, smallest, second;

smallest = second = 9999;

for(i = 0; i < 5; i++)
{
if(arr[i] < smallest)
{
second = smallest;
smallest = arr[i];
}
else if(arr[i] < second && arr[i] != smallest)
{
second = arr[i];
}
}

printf("Second Smallest = %d", second);

return 0;
}
```

Output

Second Smallest = 12

43. Merge Two Arrays

Program

\#include \<stdio.h>

```
int main()
{
int arr1[] = {10, 20, 30};
int arr2[] = {40, 50};
int arr3[5];
```

```
int i;

for(i = 0; i < 3; i++)
arr3[i] = arr1[i];

for(i = 0; i < 2; i++)
arr3[i + 3] = arr2[i];

for(i = 0; i < 5; i++)
printf("%d ", arr3[i]);

return 0;
}
```

Output

10 20 30 40 50

44. Count Frequency of Elements in an Array

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {1, 2, 2, 3, 1};
int i, count;

for(i = 0; i < 5; i++)
{
count = 0;

for(int j = 0; j < 5; j++)
{
if(arr[i] == arr[j])
count++;
}

printf("%d occurs %d times\n", arr[i], count);
}

return 0;
}
```

Output

1 occurs 2 times

2 occurs 2 times

2 occurs 2 times

3 occurs 1 times

1 occurs 2 times

45. Remove Duplicate Elements from an Array

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {1, 2, 2, 3, 4};
int i;
```

```
printf("Array without duplicates: ");

for(i = 0; i < 5; i++)
{
if(i == 0 || arr[i] != arr[i - 1])
printf("%d ", arr[i]);
}

return 0;
}
```

Output

Array without duplicates: 1 2 3 4

String Programs

46. Find the Length of a String

\#include \<stdio.h>

\#include \<string.h>

```
int main()
{
char str[] = "SkillovaHub";

printf("Length = %lu", strlen(str));

return 0;
}
```

Output

Length = 11

47. Copy One String to Another

Program

\#include \<stdio.h>

\#include \<string.h>

```
int main()
{
char str1[] = "SkillovaHub";
char str2[20];

strcpy(str2, str1);

printf("%s", str2);

return 0;
}
```

Output

SkillovaHub

48. Concatenate Two Strings

Program

\#include \<stdio.h>

\#include \<string.h>

```
int main()
{
char str1[30] = "Hello ";
char str2[] = "World";

strcat(str1, str2);

printf("%s", str1);

return 0;
}
```

Output

Hello World

49. Compare Two Strings

Program

\#include \<stdio.h>

\#include \<string.h>

```
int main()
{
char str1[] = "Apple";
char str2[] = "Apple";

if(strcmp(str1, str2) == 0)
printf("Strings are Equal");
else
printf("Strings are Not Equal");

return 0;
}
```

Output

Strings are Equal

50. Reverse a String

Program

\#include \<stdio.h>

\#include \<string.h>

```
int main()
{
char str[] = "Hello";
int i;

for(i = strlen(str) - 1; i >= 0; i-)
printf("%c", str[i]);

return 0;
}
```

Output

olleH

51. Check Palindrome String

Program

\#include \<stdio.h>

\#include \<string.h>

```
int main()
{
char str[] = "madam";
int i, len, flag = 1;

len = strlen(str);

for(i = 0; i < len / 2; i++)
{
if(str[i] != str[len - i - 1])
{
flag = 0;
break;
}
}

if(flag)
printf("Palindrome String");
else
printf("Not a Palindrome String");

return 0;
}
```

Output

Palindrome String

52. Count Vowels and Consonants

Program

\#include \<stdio.h>

```
int main()
{
char str[] = "SkillovaHub";
int i = 0, vowels = 0, consonants = 0;
char ch;

while(str[i] != '\0')
{
ch = str[i];

if(ch >= 'A' && ch <= 'Z')
ch = ch + 32;

if((ch >= 'a' && ch <= 'z'))
{
if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
vowels++;
else
consonants++;
}

i++;
}

printf("Vowels = %d\n", vowels);
```

```
printf("Consonants = %d", consonants);

return 0;
}
```

Output

Vowels = 4

Consonants = 7

53. Convert String to Uppercase

Program

\#include \<stdio.h>

\#include \<ctype.h>

```
int main()
{
char str[] = "skillovahub";
int i;

for(i = 0; str[i] != '\0'; i++)
str[i] = toupper(str[i]);

printf("%s", str);

return 0;
}
```

Output

SKILLOVAHUB

54. Convert String to Lowercase

Program

\#include \<stdio.h>

\#include \<ctype.h>

```
int main()
{
char str[] = "SKILLOVAHUB";
int i;

for(i = 0; str[i] != '\0'; i++)
str[i] = tolower(str[i]);

printf("%s", str);

return 0;
}
```

Output

skillovahub

55. Count Words in a String

Program

\#include \<stdio.h>

```
int main()
{
char str[] = "Welcome to SkillovaHub";
int i, words = 1;

for(i = 0; str[i] != '\0'; i++)
{
if(str[i] == ' ')
words++;
}

printf("Words = %d", words);

return 0;
}
```

Output

Words = 3

56. Count Characters in a String

Program

\#include \<stdio.h>

```
int main()
{
char str[] = "SkillovaHub";
int count = 0;

while(str[count] != '\0')
count++;

printf("Characters = %d", count);

return 0;
}
```

Output

Characters = 11

57. Remove Spaces from a String

Program

\#include \<stdio.h>

```
int main()
{
char str[] = "Hello World";
int i;

for(i = 0; str[i] != '\0'; i++)
{
if(str[i] != ' ')
printf("%c", str[i]);
}

return 0;
}
```

Output

HelloWorld

58. Find Frequency of Characters in a String

Program

\#include \<stdio.h>

```
int main()
{
char str[] = "banana";
char ch = 'a';
int i, count = 0;

for(i = 0; str[i] != '\0'; i++)
{
if(str[i] == ch)
count++;
}

printf("Frequency of '%c' = %d", ch, count);

return 0;
}
```

Output

Frequency of 'a' = 3

59. Check Anagram Strings

Program

\#include \<stdio.h>

\#include \<string.h>

```
int main()
{
char str1[] = "listen";
char str2[] = "listen";

if(strcmp(str1, str2) == 0)
printf("Anagram");
else
printf("Not an Anagram");

return 0;
}
```

Output

Anagram

60. Find Duplicate Characters in a String

Program

\#include \<stdio.h>

```
int main()
{
char str[] = "programming";
int i, j;

printf("Duplicate Characters: ");

for(i = 0; str[i] != '\0'; i++)
```

```
{
for(j = i + 1; str[j] != '\0'; j++)
{
if(str[i] == str[j])
{
printf("%c ", str[i]);
break;
}
}
}

return 0;
}
```

Output

Duplicate Characters: r g m

Function Programs

61. Function Without Arguments and Without Return Value

Program

\#include \<stdio.h>

```
void display()
{
printf("Welcome to C Programming");
}

int main()
{
display();

return 0;
}
```

Output

Welcome to C Programming

62. Function With Arguments and Without Return Value

Program

\#include \<stdio.h>

```
void add(int a, int b)
{
printf("Sum = %d", a + b);
}

int main()
{
add(10, 20);

return 0;
}
```

Output

Sum = 30

63. Function With Arguments and Return Value

Program

\#include \<stdio.h>

```
int add(int a, int b)
{
return a + b;
}

int main()
{
printf("Sum = %d", add(10, 20));

return 0;
}
```

Output

Sum = 30

64. Recursive Factorial Program

Program

\#include \<stdio.h>

```
int factorial(int n)
{
if(n == 0 || n == 1)
return 1;

return n * factorial(n - 1);
}

int main()
{
printf("%d", factorial(5));

return 0;
}
```

Output

120

65. Recursive Fibonacci Series

Program

\#include \<stdio.h>

```
int fibonacci(int n)
{
if(n <= 1)
return n;

return fibonacci(n - 1) + fibonacci(n - 2);
}

int main()
{
int i;

for(i = 0; i < 10; i++)
printf("%d ", fibonacci(i));

return 0;
```

```
}
```

Output

0 1 1 2 3 5 8 13 21 34

66. Call by Value

Program

\#include \<stdio.h>

```
void swap(int a, int b)
{
int temp;
temp = a;
a = b;
b = temp;

printf("Inside Function:\n");
printf("a = %d b = %d\n", a, b);
}

int main()
{
int a = 10, b = 20;

swap(a, b);

printf("Outside Function:\n");
printf("a = %d b = %d", a, b);

return 0;
}
```

Output

Inside Function:

```
a = 20 b = 10
Outside Function:
a = 10 b = 20
```

67. Call by Reference

Program

\#include \<stdio.h>

```
void swap(int *a, int *b)
{
int temp;

temp = *a;
*a = *b;
*b = temp;
}

int main()
{
int a = 10, b = 20;

swap(&a, &b);

printf("a = %d\n", a);
printf("b = %d", b);
```

```
return 0;
}
```

Output

```
a = 20
b = 10
```

68. Find Maximum Using Function

Program

\#include \<stdio.h>

```
int maximum(int a, int b)
{
if(a > b)
return a;
else
return b;
}

int main()
{
printf("Maximum = %d", maximum(25, 18));

return 0;
}
```

Output

Maximum = 25

69. Find Minimum Using Function

Program

\#include \<stdio.h>

```
int minimum(int a, int b)
{
if(a < b)
return a;
else
return b;
}

int main()
{
printf("Minimum = %d", minimum(25, 18));

return 0;
}
```

Output

Minimum = 18

70. Simple Calculator Using Functions

Program

\#include \<stdio.h>

```
int add(int a, int b)
{
```

```
return a + b;
}

int main()
{
printf("Sum = %d", add(15, 10));

return 0;
}
```

Output

Sum = 25

Pointer Programs

71. Display Pointer Value and Address

Program

\#include \<stdio.h>

```
int main()
{
int num = 10;
int *ptr = &num;

printf("Value = %d\n", *ptr);
printf("Address = %p", (void *)ptr);

return 0;
}
```

Output

Value = 10

Address = 0x...

Note: The memory address (`0x...`) will be different on every system.

72. Pointer Arithmetic

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {10, 20, 30};
int *ptr = arr;

printf("%d\n", *ptr);

ptr++;

printf("%d", *ptr);

return 0;
}
```

Output

10

20

73. Swap Two Numbers Using Pointers

Program

\#include \<stdio.h>

```
int main()
{
int a = 10, b = 20;
int *p1 = &a;
int *p2 = &b;
int temp;

temp = *p1;
*p1 = *p2;
*p2 = temp;

printf("a = %d\n", a);
printf("b = %d", b);

return 0;
}
```

Output

```
a = 20
b = 10
```

74. Find the Largest Number Using Pointers

Program

\#include \<stdio.h>

```
int main()
{
int a = 25, b = 40;
int *p1 = &a;
int *p2 = &b;

if(*p1 > *p2)
printf("Largest = %d", *p1);
else
printf("Largest = %d", *p2);

return 0;
}
```

Output

Largest = 40

75. Sum of Array Elements Using Pointers

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {10, 20, 30, 40, 50};
int *ptr = arr;
int i, sum = 0;

for(i = 0; i < 5; i++)
```

```
sum += *(ptr + i);

printf("Sum = %d", sum);

return 0;
}
```

Output

Sum = 150

76. Print Array Using Pointers

Program

\#include \<stdio.h>

```
int main()
{
int arr[] = {10, 20, 30, 40, 50};
int *ptr = arr;
int i;

for(i = 0; i < 5; i++)
printf("%d ", *(ptr + i));

return 0;
}
```

Output

10 20 30 40 50

77. String Traversal Using Pointers

Program

\#include \<stdio.h>

```
int main()
{
char str[] = "SkillovaHub";
char *ptr = str;

while(*ptr != '\0')
{
printf("%c", *ptr);
ptr++;
}

return 0;
}
```

Output

SkillovaHub

78. Pointer to Pointer

Program

\#include \<stdio.h>

```
int main()
{
int num = 10;
```

```
int *ptr = &num;
int **pptr = &ptr;

printf("Value = %d", **pptr);

return 0;
}
```

Output

Value = 10

79. Compare Two Pointer Addresses

Program

\#include \<stdio.h>

```
int main()
{
int a = 10, b = 20;
int *p1 = &a;
int *p2 = &b;

if(p1 == p2)
printf("Pointers are Equal");
else
printf("Pointers are Different");

return 0;
}
```

Output

Pointers are Different

80. Function Using Pointers

Program

\#include \<stdio.h>

```
void square(int *n)
{
*n = (*n) * (*n);
}

int main()
{
int num = 5;

square(&num);

printf("Square = %d", num);

return 0;
}
```

Output

Square = 25

File Handling Programs

81. Create a File

Program

\#include \<stdio.h>

```
int main()
{
FILE *fp;

fp = fopen("sample.txt", "w");

if(fp == NULL)
printf("File could not be created.");
else
{
printf("File created successfully.");
fclose(fp);
}

return 0;
}
```

Output

File created successfully.

82. Write Data to a File

Program

\#include \<stdio.h>

```
int main()
{
FILE *fp;

fp = fopen("sample.txt", "w");

fprintf(fp, "Welcome to SkillovaHub");

fclose(fp);

printf("Data written successfully.");

return 0;
}
```

Output

Data written successfully.

83. Read Data from a File

Program

\#include \<stdio.h>

```
int main()
{
FILE *fp;
char ch;

fp = fopen("sample.txt", "r");

while((ch = fgetc(fp)) != EOF)
putchar(ch);

fclose(fp);
```

```
return 0;
}
```

Output

Welcome to SkillovaHub

84. Append Data to a File

Program

\#include \<stdio.h>

```
int main()
{
FILE *fp;

fp = fopen("sample.txt", "a");

fprintf(fp, "\nLearn C Programming");

fclose(fp);

printf("Data appended successfully.");

return 0;
}
```

Output

Data appended successfully.

85. Copy File Contents

Program

\#include \<stdio.h>

```
int main()
{
FILE *source, *destination;
char ch;

source = fopen("sample.txt", "r");
destination = fopen("copy.txt", "w");

while((ch = fgetc(source)) != EOF)
fputc(ch, destination);

fclose(source);
fclose(destination);

printf("File copied successfully.");

return 0;
}
```

Output

File copied successfully.

86. Count Characters in a File

Program

\#include \<stdio.h>

```
int main()
{
FILE *fp;
char ch;
int count = 0;

fp = fopen("sample.txt", "r");

while((ch = fgetc(fp)) != EOF)
count++;

fclose(fp);

printf("Characters = %d", count);

return 0;
}
```

Output

Characters = 35

87. Count Words in a File

Program

\#include \<stdio.h>

```
int main()
{
FILE *fp;
char ch;
int words = 0;

fp = fopen("sample.txt", "r");

while((ch = fgetc(fp)) != EOF)
{
if(ch == ' ' || ch == '\n')
words++;
}

fclose(fp);

printf("Words = %d", words + 1);

return 0;
}
```

Output

Words = 5

88. Count Lines in a File

Program

\#include \<stdio.h>

```
int main()
{
FILE *fp;
char ch;
int lines = 1;
```

```
fp = fopen("sample.txt", "r");

while((ch = fgetc(fp)) != EOF)
{
if(ch == '\n')
lines++;
}

fclose(fp);

printf("Lines = %d", lines);

return 0;
}
```

Output

Lines = 2

89. Rename a File

Program

\#include \<stdio.h>

```
int main()
{
if(rename("sample.txt", "newfile.txt") == 0)
printf("File renamed successfully.");
else
printf("Unable to rename file.");

return 0;
}
```

Output

File renamed successfully.

90. Delete a File

Program

\#include \<stdio.h>

```
int main()
{
if(remove("newfile.txt") == 0)
printf("File deleted successfully.");
else
printf("Unable to delete file.");

return 0;
}
```

Output

File deleted successfully.

Interview Programs

91. Menu-Driven Calculator

Program

\#include \<stdio.h>

```
int main()
{
int choice;
int a = 20, b = 10;

choice = 1;

switch(choice)
{
case 1:
printf("Addition = %d", a + b);
break;

case 2:
printf("Subtraction = %d", a - b);
break;

case 3:
printf("Multiplication = %d", a * b);
break;

case 4:
printf("Division = %d", a / b);
break;

default:
printf("Invalid  Choice");
}

return 0;
}
```

Output

Addition = 30

92. Student Record Using Structure

Program

\#include \<stdio.h>

```
struct Student
{
int id;
char name[20];
};

int main()
{
struct Student s = {101, "Rahul"};

printf("ID = %d\n", s.id);
printf("Name = %s", s.name);

return 0;
}
```

Output

ID = 101

Name = Rahul

93. Employee Record Using Structure

Program

\#include \<stdio.h>

```
struct Employee
{
int id;
float salary;
};

int main()
{
struct Employee e = {101, 35000};

printf("ID = %d\n", e.id);
printf("Salary = %.2f", e.salary);

return 0;
}
```

Output

ID = 101

Salary = 35000.00

94. Matrix Addition

Program

\#include \<stdio.h>

```
int main()
{
int a[2][2]={{1,2},{3,4}};
int b[2][2]={{5,6},{7,8}};
int c[2][2];
int i,j;

for(i=0;i<2;i++)
{
for(j=0;j<2;j++)
c[i][j]=a[i][j]+b[i][j];
}

for(i=0;i<2;i++)
{
for(j=0;j<2;j++)
printf("%d ",c[i][j]);
printf("\n");
}

return 0;
}
```

Output

6 8

10 12

95. Matrix Multiplication

Program

\#include \<stdio.h>

```
int main()
{
int a[2][2]={{1,2},{3,4}};
int b[2][2]={{5,6},{7,8}};
int c[2][2];
int i,j,k;

for(i=0;i<2;i++)
{
for(j=0;j<2;j++)
{
c[i][j]=0;
for(k=0;k<2;k++)
c[i][j]+=a[i][k]*b[k][j];
}
}

for(i=0;i<2;i++)
{
for(j=0;j<2;j++)
printf("%d ",c[i][j]);
printf("\n");
}

return 0;
}
```

Output

19 22

43 50

96. Transpose of Matrix

Program

\#include \<stdio.h>

```
int main()
{
int a[2][2]={{1,2},{3,4}};
int i,j;

for(i=0;i<2;i++)
{
for(j=0;j<2;j++)
printf("%d ",a[j][i]);

printf("\n");
}

return 0;
}
```

Output

1 3

2 4

97. Linear Search

Program

\#include \<stdio.h>

```
int main()
{
int arr[]={10,20,30,40,50};
int key=30,i;

for(i=0;i<5;i++)
{
if(arr[i]==key)
{
printf("Element Found");
return 0;
}
}

printf("Element Not Found");

return 0;
}
```

Output

Element Found

98. Binary Search

Program

\#include \<stdio.h>

```
int main()
{
int arr[]={10,20,30,40,50};
int low=0,high=4,mid,key=40;

while(low<=high)
{
mid=(low+high)/2;

if(arr[mid]==key)
{
printf("Element Found");
return 0;
}
else if(arr[mid]<key)
low=mid+1;
else
high=mid-1;
}

printf("Element Not Found");

return 0;
}
```

Output

Element Found

99. Bubble Sort

Program

\#include \<stdio.h>

```
int main()
{
int arr[]={5,3,4,1,2};
int i,j,temp;

for(i=0;i<5;i++)
{
for(j=0;j<4-i;j++)
{
if(arr[j]>arr[j+1])
{
temp=arr[j];
arr[j]=arr[j+1];
arr[j+1]=temp;
}
}
}

for(i=0;i<5;i++)
printf("%d ",arr[i]);

return 0;
}
```

Output

1 2 3 4 5

100. Selection Sort

Program

\#include \<stdio.h>

```
int main()
{
int arr[]={64,25,12,22,11};
int i,j,min,temp;

for(i=0;i<5;i++)
{
min=i;

for(j=i+1;j<5;j++)
{
if(arr[j]<arr[min])
min=j;
}

temp=arr[min];
arr[min]=arr[i];
arr[i]=temp;
}

for(i=0;i<5;i++)
printf("%d ",arr[i]);

return 0;
}
```

Output

11 12 22 25 64