**Reverse an Integer**

#include <stdio.h>

int main() {

int n, reverse = 0, remainder;

printf("Enter an integer: ");

scanf("%d", &n);

while (n != 0) {

remainder = n % 10;

reverse = reverse \* 10 + remainder;

n /= 10;

}

printf("Reversed number = %d", reverse);

return 0;

}

[Run Code](https://www.programiz.com/c-programming/online-compiler)

**Output**

Enter an integer: 2345

Reversed number = 5432

Reverse a string

**Program1.c**

1. #include <stdio.h>
2. #include <string.h>
3. **int** main()
4. {
5. **char** str[40]; // declare the size of character string
6. printf (" \n Enter a string to be reversed: ");
7. scanf ("%s", str);
9. // use strrev() function to reverse a string
10. printf (" \n After the reverse of a string: %s ", strrev(str));
11. **return** 0;
12. }

**Output**

Enter a string to be reversed: AMBULANCE

After the reverse of a string: ECNALUBMA

Palindrome a number

1. #include<stdio.h>
2. **int** main()
3. {
4. **int** n,r,sum=0,temp;
5. printf("enter the number=");
6. scanf("%d",&n);
7. temp=n;
8. **while**(n>0)
9. {
10. r=n%10;
11. sum=(sum\*10)+r;
12. n=n/10;
13. }
14. **if**(temp==sum)
15. printf("palindrome number ");
16. **else**
17. printf("not palindrome");
18. **return** 0;
19. }

**Output:**

enter the number=151

palindrome number

enter the number=5621

not palindrome number

Palindrome a string

|  |
| --- |
| #include <stdio.h>  #include <string.h>    **int** main()  {  **char** str[] = { "abbba" };      **int** l = 0;  **int** h = **strlen**(str) - 1;      **while** (h > l) {  **if** (str[l++] != str[h--]) {  **printf**("%s is not a palindrome\n", str);  **return** 0;          }      }    **printf**("%s is a palindrome\n", str);    **return** 0;  } |

**Output**

abbba is a palindrome

**Program to Check Prime Number**

#include <stdio.h>

int main() {

int n, i, flag = 0;

printf("Enter a positive integer: ");

scanf("%d", &n);

// 0 and 1 are not prime numbers

// change flag to 1 for non-prime number

if (n == 0 || n == 1)

flag = 1;

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

// if n is divisible by i, then n is not prime

// change flag to 1 for non-prime number

if (n % i == 0) {

flag = 1;

break;

}

}

// flag is 0 for prime numbers

if (flag == 0)

printf("%d is a prime number.", n);

else

printf("%d is not a prime number.", n);

return 0;

}

[Run Code](https://www.programiz.com/c-programming/online-compiler)

**Output**

Enter a positive integer: 29

29 is a prime number.

**Check Armstrong Number of three digits**

#include <stdio.h>

int main() {

int num, originalNum, remainder, result = 0;

printf("Enter a three-digit integer: ");

scanf("%d", &num);

originalNum = num;

while (originalNum != 0) {

// remainder contains the last digit

remainder = originalNum % 10;

result += remainder \* remainder \* remainder;

// removing last digit from the orignal number

originalNum /= 10;

}

if (result == num)

printf("%d is an Armstrong number.", num);

else

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

return 0;

}

[Run Code](https://www.programiz.com/c-programming/online-compiler)

**Output**

Enter a three-digit integer: 371

371 is an Armstrong number.

Fibonacci Series in C without recursion

Let's see the fibonacci series program in c without recursion.

1. #include<stdio.h>
2. **int** main()
3. {
4. **int** n1=0,n2=1,n3,i,number;
5. printf("Enter the number of elements:");
6. scanf("%d",&number);
7. printf("\n%d %d",n1,n2);//printing 0 and 1
8. **for**(i=2;i<number;++i)//loop starts from 2 because 0 and 1 are already printed
9. {
10. n3=n1+n2;
11. printf(" %d",n3);
12. n1=n2;
13. n2=n3;
14. }
15. **return** 0;
16. }

**Output:**

Enter the number of elements:15

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

**Factorial of a Number**

#include <stdio.h>

int main() {

int n, i;

unsigned long long fact = 1;

printf("Enter an integer: ");

scanf("%d", &n);

// shows error if the user enters a negative integer

if (n < 0)

printf("Error! Factorial of a negative number doesn't exist.");

else {

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

fact \*= i;

}

printf("Factorial of %d = %llu", n, fact);

}

return 0;

}

**Output**

Enter an integer: 10

Factorial of 10 = 3628800

Find duplicate of an arrays

1. #include <stdio.h>
3. **int** main()
4. {
5. //Initialize array
6. **int** arr[] = {1, 2, 3, 4, 2, 7, 8, 8, 3};
8. //Calculate length of array arr
9. **int** length = **sizeof**(arr)/**sizeof**(arr[0]);
11. printf("Duplicate elements in given array: \n");
12. //Searches for duplicate element
13. **for**(**int** i = 0; i < length; i++) {
14. **for**(**int** j = i + 1; j < length; j++) {
15. **if**(arr[i] == arr[j])
16. printf("%d\n", arr[j]);
17. }
18. }
19. **return** 0;
20. }

**Output:**

Duplicate elements in given array:

2

3

8

### C Program to find largest and smallest element in array

**#include <stdio.h>**

//Calculate array size

**#define ARRAY\_SIZE(a) sizeof(a)/sizeof(a[0])**

**int** main()

{

**int** arr[] = {3, 18, 10, 4, 2, 22, 150};

**int** i, small, large;

const **int** N = ARRAY\_SIZE(arr);

small = arr[0];//Assume first element is smallest

large = arr[0];//Assume first element is largest

//iterate through the array

**for** (i = 1; i < N; i++)

{

**if** (arr[i] < small)

{

small = arr[i];

}

**if** (arr[i] > large)

{

large = arr[i];

}

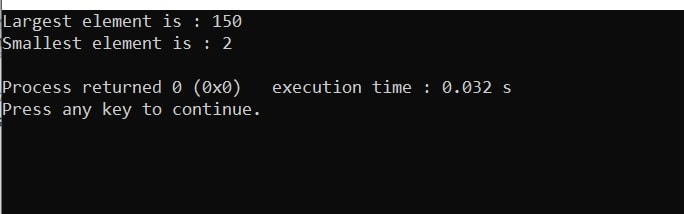
}

printf("Largest element is : %d\n", large);

printf("Smallest element is : %d\n", small);

**return** 0;

}



Reverse an array

#include <stdio.h>

#include <stdlib.h>

#define n 6

int main(){

int arr[n] = {9, 8, 7, 2, 4, 3};

int temp;

for(int i = 0; i<n/2; i++){

temp = arr[i];

arr[i] = arr[n-i-1];

arr[n-i-1] = temp;

}

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

printf("%d,", arr[i]);

}

}

## Input

6, 9, 8, 7, 2, 4, 3

## Output

3,4,2,7,8,9,

Sort of an arrays

#include <stdio.h>

void main (){

   int num[20];

   int i, j, a, n;

   printf("enter number of elements in an array  
");

   scanf("%d", &n);

   printf("Enter the elements  
");

   for (i = 0; i < n; ++i)

      scanf("%d", &num[i]);

   for (i = 0; i < n; ++i){

      for (j = i + 1; j < n; ++j){

         if (num[i] > num[j]){

            a = num[i];

            num[i] = num[j];

            num[j] = a;

         }

      }

   }

   printf("The numbers in ascending order is:  
");

   for (i = 0; i < n; ++i){

      printf("%d  
", num[i]);

   }

}

## Output

When the above program is executed, it produces the following result −

enter number of elements in an array

5

Enter the elements

12

23

89

11

22

The numbers in ascending order is:

11

12

22

23

89