

Table of Contents

Write a recursive function to find the factorial of a given number.	2
Write a recursive function to print the 'n' fibonacci series numbers.	4
Write a recursive function to find the sum of digits of a given number.	6
Write a recursive function to reverse the given number.	8
Write a recursive function to that displays all the proper divisors of a given number except that and returns their sum.	10
Write a recursive function that displays a positive integer in words. For ex: if the integer is 3412 then it is displayed as three four one two.	12
Write a recursive function to print first 100 prime numbers.	15
Write a recursive function to print the palindrome numbers in a given numbers.	17
Write a recursive function that finds whether a number is perfect or not.	19
Write a recursive function to find the largest element in a given Unsorted array.	21
Write a recursive function to reverse the bits of a given number.	23
Write a recursive function to reverse the elements of a given array.	25
Write a recursive function to reverse the string. (Note : not just reverse printing character by character)	27

Write a recursive function to find the factorial of a given number.

```
#include<stdio.h>

main()
{
    int i,num,ans;

    printf("Enter Number : \n");
    scanf("%d",&num);

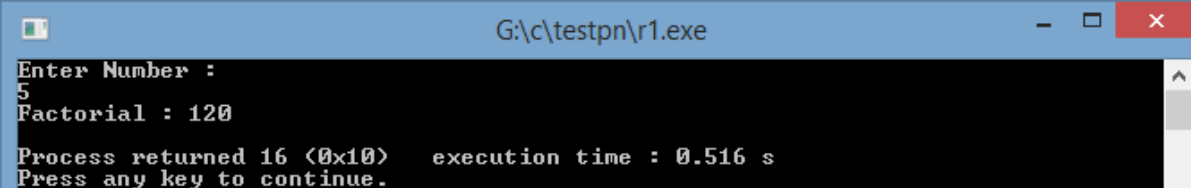
    ans=my_fact_rev(num);

    printf("Factorial : %d\n",ans);
}

int my_fact_rev(int num)
{
    if(num)
    {
        return num *my_fact_rev(num-1);
    }
    else
        return 1;
}
```

```
int tmy_fact_rev(int num)
{
    static int i=1,t=1;

    if(i <= num)
    {
        t = t * i;
        i++;
        my_fact_rev(num);
        return t;
    }
    else
        return t;
}
```



```
G:\c\testpn\r1.exe
Enter Number :
5
Factorial : 120
Process returned 16 (0x10)   execution time : 0.516 s
Press any key to continue.
```

Write a recursive function to print the 'n' fibonacci series numbers.

```
#include<stdio.h>

void my_fibbo_rec(int);

main()
{
    int num;

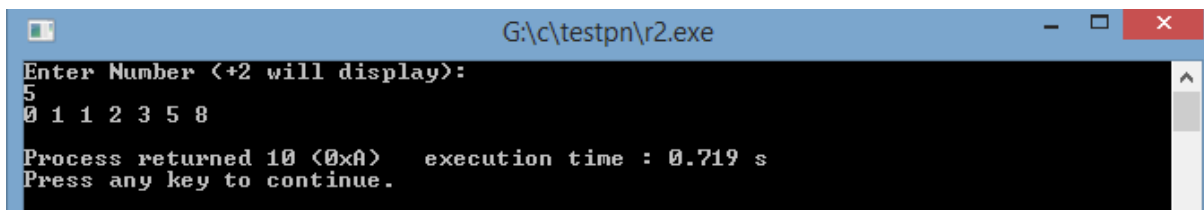
    printf("Enter Number (+2 will display): \n");
    scanf("%d",&num);

    printf("0 1 ");
    my_fibbo_rec(num);
    printf("\n");
}

void my_fibbo_rec(int n)
{
    static int i=1,j=0,k=1,l;

    if(i<=n)
    {
        l=j+k;
        j=k;
        k=l;
    }
}
```

```
    printf("%d ",l);  
  
    i++;  
  
    my_fibbo_rec(n);  
  
    }  
  
}
```



```
G:\c\testpn\r2.exe  
Enter Number (<+2 will display>:  
5  
0 1 1 2 3 5 8  
Process returned 10 (0xA)   execution time : 0.719 s  
Press any key to continue.
```

Write a recursive function to find the sum of digits of a given number.

```
#include<stdio.h>

int my_sum_rec(int);

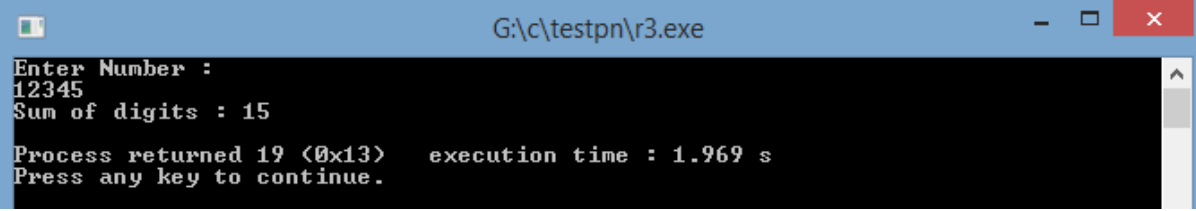
main()
{
    int num,ans;

    printf("Enter Number : \n");
    scanf("%d",&num);

    ans=my_sum_rec(num);
    printf("Sum of digits : %d\n",ans);
}

int my_sum_rec(int num)
{
    static int s=0;
    if(num)
    {
        s = s + num % 10;
        num=num/10;
        my_sum_rec(num);
    }
}
```

```
    return s;  
}  
}
```



```
Enter Number :  
12345  
Sum of digits : 15  
Process returned 19 (0x13)   execution time : 1.969 s  
Press any key to continue.
```

Write a recursive function to reverse the given number.

```
#include<stdio.h>

int my_sum_rec(int);

main()
{
    int num,ans;

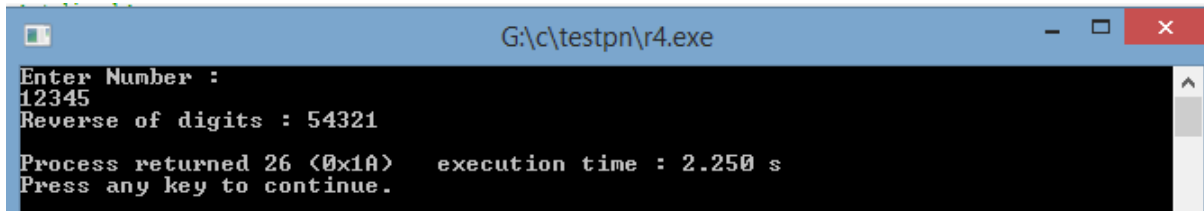
    printf("Enter Number : \n");
    scanf("%d",&num);

    ans=my_sum_rec(num);
    printf("Reverse of digits : %d\n",ans);
}

int my_sum_rec(int num)
{
    static int s=0;
    if(num)
    {
        s = 10*s + num % 10;
        num=num/10;
        my_sum_rec(num);
    }
}
```



```
    return s;  
}  
}
```



```
G:\c\testpn\r4.exe  
Enter Number :  
12345  
Reverse of digits : 54321  
Process returned 26 (0x1A)   execution time : 2.250 s  
Press any key to continue.
```

Write a recursive function to that displays all the proper divisors of a given number except that and returns their sum.

Ex: 1,3,5,9,15 & 45 are the proper divisors of 45.

$$\text{sum} = 1+3+5+9+15$$
$$\text{sum} = 33$$

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

```
int my_per_rec(int);
```

```
main()
```

```
{
```

```
    int num,ans;
```

```
    printf("Enter number : \n");
```

```
    scanf("%d",&num);
```

```
    ans = my_per_rec(num);
```

```
    printf("\nTotal : %d\n",ans);
```

```
}
```

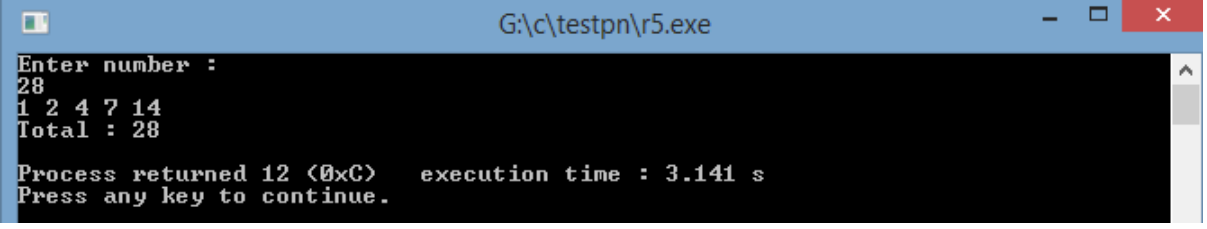
```
int my_per_rec(int num)
```

```
{
```

```
    static int i=1,s=0;
```

```
    if(i < num)
```

```
{  
    if(num % i == 0)  
    {  
        s = s + i;  
        printf("%d ",i);  
    }  
    i = i + 1;  
    my_per_rec(num);  
    return s;  
}  
}
```



```
G:\c\testpn\r5.exe  
Enter number :  
28  
1 2 4 7 14  
Total : 28  
Process returned 12 (0xC)   execution time : 3.141 s  
Press any key to continue.
```

Write a recursive function that displays a positive integer in words.
For ex: if the integer is 3412 then it is displayed as three four one two.

```
#include<stdio.h>

void my_call(int);

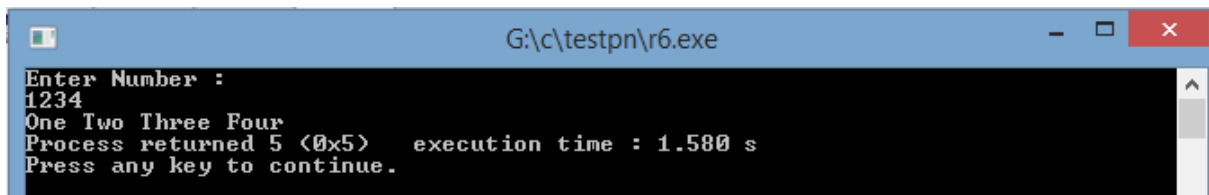
main()
{
    int s,n=0;
    int i,j,k,m;

    printf("Enter Number : \n");
    scanf("%d",&s);
    my_call(s);
}

void my_call(int num)
{
    int i;

    if(num)
    {
        i = num%10;
        my_call(num/10);
        switch(i)
```

```
{  
  
    case 0:printf("Zero ");  
    break;  
  
    case 8:printf("Eight ");  
    break;  
  
    case 9:printf("Nine ");  
    break;  
  
    case 7:printf("Seven ");  
    break;  
  
    case 1:printf("One ");  
    break;  
  
    case 2:printf("Two ");  
    break;  
  
    case 3:printf("Three ");  
    break;  
  
    case 4:printf("Four ");  
    break;  
  
    case 5:printf("Five ");  
    break;  
  
    case 6:printf("Six ");  
    break;  
  
}  
  
}
```



```
Enter Number :  
1234  
One Two Three Four  
Process returned 5 (0x5) execution time : 1.580 s  
Press any key to continue.
```

Write a recursive function to print first 100 prime numbers.

```
#include<stdio.h>

int my_prime_rec(int,int);

main()
{
    int a,i,j=2,c=0;

    for(i=1;c<100;i++)
    {
        a = my_prime_rec(i,j);
        if(a==i)
        {
            printf("%d ",i);
            c++;
        }
    }
    printf("\n");
}

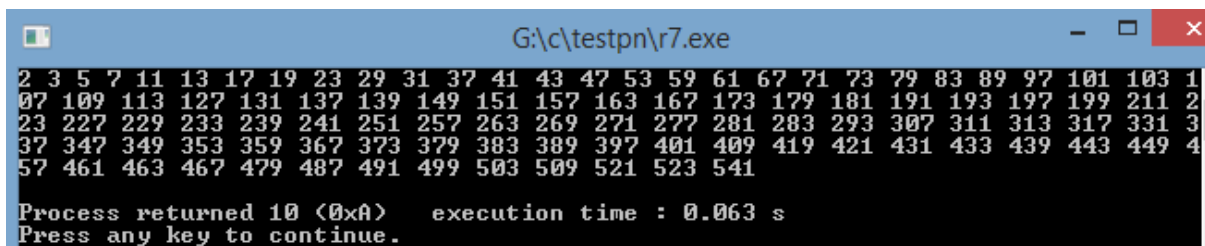
int my_prime_rec(int n,int j)
{
    //static int i=2,t=0;
```

```

    if(j<n)
    {
        if(n % j == 0)
            return j;

        j=j+1;
        my_prime_rec(n,j);
    }
}

```



```

G:\c\testpn\r7.exe
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223 227 229 233 239 241 251 257 263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 359 367 373 379 383 389 397 401 409 419 421 431 433 439 443 449 457 461 463 467 479 487 491 499 503 509 521 523 541
Process returned 10 (0x0A) execution time : 0.063 s
Press any key to continue.

```


Write a recursive function to print the palindrome numbers in a given numbers.

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

```
int my_palin_rec(int);
```

```
main()
```

```
{
```

```
    int num,ans;
```

```
    printf("Enter Number : \n");
```

```
    scanf("%d",&num);
```

```
    ans=my_palin_rec(num);
```

```
    if(ans == num)
```

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

```
    else
```

```
        printf("Not...\n");
```

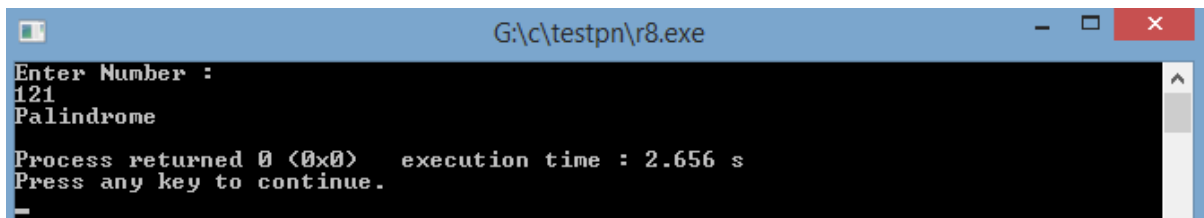
```
}
```

```
int my_palin_rec(int num)
```

```
{
```

```
    static int s=0;
```

```
    if(num)
    {
        s = 10*s + num % 10;
        num=num/10;
        my_palin_rec(num);
        return s;
    }
}
```



```
G:\c\testpn\r8.exe
Enter Number :
121
Palindrome
Process returned 0 (0x0)   execution time : 2.656 s
Press any key to continue.
_
```

Write a recursive function that finds whether a number is perfect or not.

```
#include<stdio.h>

int my_per_rec(int);

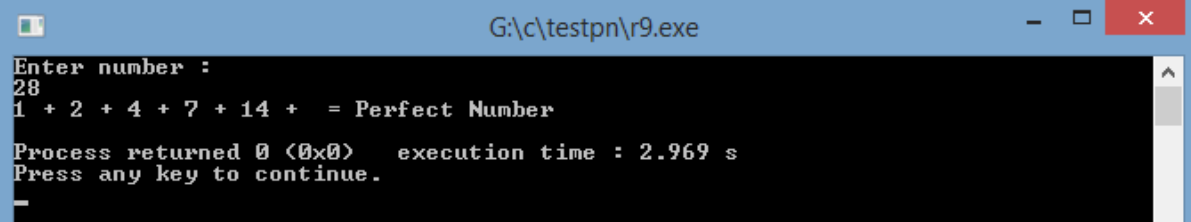
main()
{
    int num,ans;
    printf("Enter number : \n");
    scanf("%d",&num);

    ans = my_per_rec(num);
    if(ans==num)
        printf(" = Perfect Number\n");
    else
        printf("\nNot...\n");
}
```

```
int my_per_rec(int num)
{
    static int i=1,s=0,max=0;

    if(i < num)
```

```
{  
    if(num % i == 0)  
    {  
        s = s + i;  
        printf("%d + ",i);  
    }  
    i = i + 1;  
    my_per_rec(num);  
    return s;  
}  
}
```



```
G:\c\testpn\r9.exe  
Enter number :  
28  
1 + 2 + 4 + 7 + 14 + = Perfect Number  
Process returned 0 (0x0)   execution time : 2.969 s  
Press any key to continue.  
_
```

Write a recursive function to find the largest element in a given Unsorted array.

```
#include<stdio.h>

int max_rec(int*,int);

main()
{
    int a[5],i,ele,max;

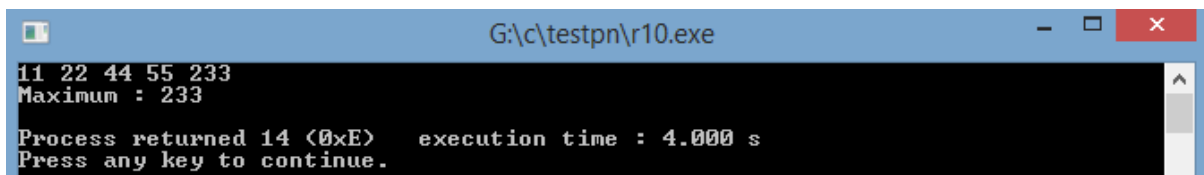
    ele = sizeof(a)/sizeof(a[0]);
    for(i=0;i<ele;i++)
    {
        scanf("%d",&a[i]);
    }

    max = max_rec(a,ele);
    printf("Maximum : %d\n",max);
}

int max_rec(int*p,int ele)
{
    static int i=1,maxx=0;

    if(i<=ele)
    {
```

```
    if((*p) > maxx)
    maxx = *p;
    i++;
    max_rec(p+1,ele);
    return maxx;
}
}
```



```
G:\c\testpn\r10.exe
11 22 44 55 233
Maximum : 233
Process returned 14 (0xE)    execution time : 4.000 s
Press any key to continue.
```

Write a recursive function to reverse the bits of a given number.

```
#include<stdio.h>

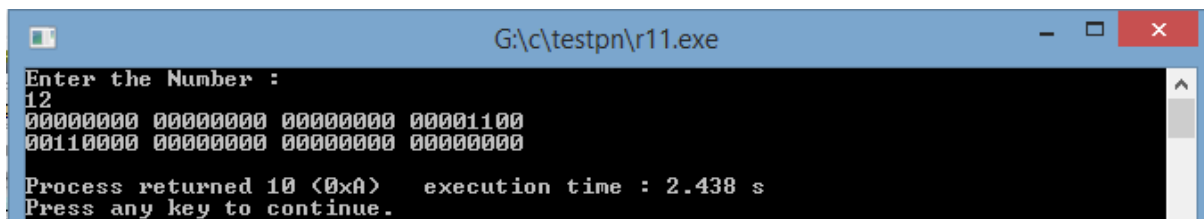
void rev(int*);

main()
{
    int num,i,k;
    printf("Enter the Number : \n");
    scanf("%d",&num);
    for(i=31;i>=0;i--)
    {
        printf("%d",num>>i&1);
        if(!(i%8))
            printf(" ");
    }
    printf("\n");
    rev(&num);
    for(i=31;i>=0;i--)
    {
        printf("%d",num>>i&1);
        if(!(i%8))
            printf(" ");
    }
    printf("\n");
}
```

```

void rev(int*p)
{
    static int i=0,j=31,k;
    if(i<j)
    {
        if((*p >> i & 1) != (*p >> j & 1))
        {
            *p = *p ^ 1 << i;
            *p = *p ^ 1 << j;
        }
        i++;
        j--;
        rev(p);
    }
}

```



```

G:\c\testpn\r11.exe
Enter the Number :
12
00000000 00000000 00000000 00001100
00110000 00000000 00000000 00000000

Process returned 10 (0xA)   execution time : 2.438 s
Press any key to continue.

```


Write a recursive function to reverse the elements of a given array.

```
#include<stdio.h>

void my_ar(int*,int*);

main()
{
    int a[5],ele,i,*s;

    ele=sizeof(a)/sizeof(a[0]);

    printf("Enter %d Elements : \n",ele);
    for(i=0;i<ele;i++)
    {
        scanf("%d",&a[i]);
    }

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

    s=a+ele-1;

    my_ar(a,s);

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

```

    {
        printf("%d ",a[i]);
    }
    printf("\n");
}

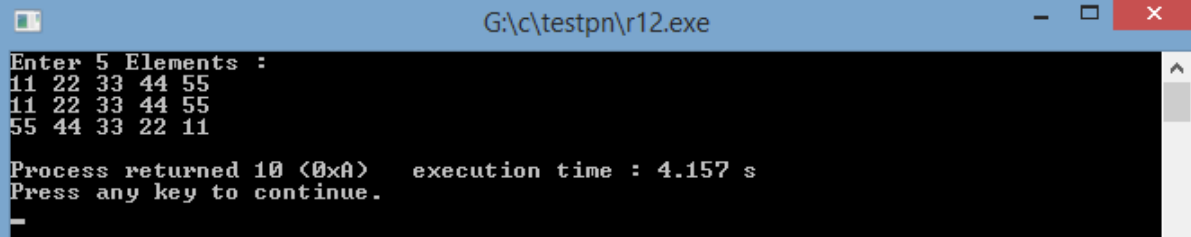
```

```

void my_ar(int *p,int *q)
{
    int t;

    if(p < q)
    {
        t=*p;
        *p=*q;
        *q=t;
        my_ar(p+1,q-1);
    }
}

```



```

G:\c\testpn\r12.exe
Enter 5 Elements :
11 22 33 44 55
11 22 33 44 55
55 44 33 22 11
Process returned 10 (0xA)   execution time : 4.157 s
Press any key to continue.
_

```

Write a recursive function to reverse the string. (Note : not just reverse printing character by character)

```
#include<stdio.h>

void str_rev(char *,char *);

main()
{
    char s[30],ch,*q;
    int i,j,len;
    printf("Enter String : \n");
    scanf("%s",s);

    printf("Originl : %s\n",s);
    for(i=0;s[i];i++);

    //len=i-1;

    q = s+i-1;

    str_rev(s,q);

    /*for(i=0,j=(strlen(s)-1);i<j;i++,j--)
    {
        ch=s[i];
        s[i]=s[j];
        s[j]=ch;
    }
    */
```

```

        s[j]=ch;
    }*/

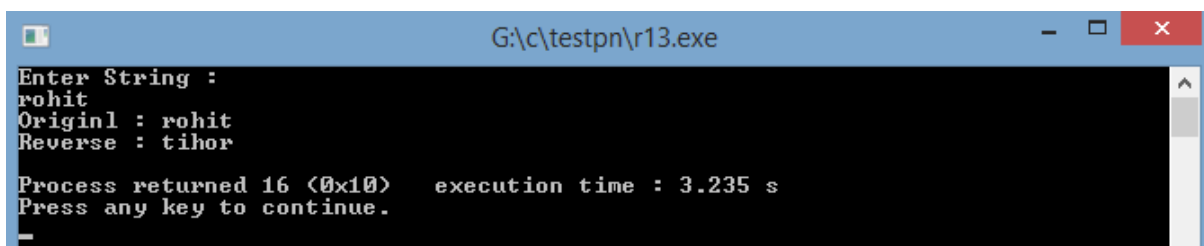
    printf("Reverse : %s\n",s);
}

```

```

void str_rev(char*p, char *q)
{
    char c;
    if(p<q)
    {
        c = *p;
        *p = *q;
        *q = c;
        str_rev(p+1,q-1);
    }
}

```



```

G:\c\testpn\r13.exe
Enter String :
rohit
Original : rohit
Reverse : tihor
Process returned 16 (0x10)   execution time : 3.235 s
Press any key to continue.
_

```