

พื้นฐานภาษา C

1. การแสดงข้อความ

```
#include<stdio.h>
int main()
{
    printf("message");
    return 0;
}
```

2. การแสดงข้อความจากตัวแปร

```
#include<stdio.h>
int main()
{
    int      a = 100;
    float    b = 2.2;
    char     c = 'c';
    printf("message %d %f %c",a,b,c);
    return 0;
}
```

3. การรับข้อมูล

```
#include<stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    printf("%d",a);
    return 0;
}
```

4. ตัวแปร Local ใช้ได้เฉพาะภายในฟังก์ชัน

```
#include<stdio.h>
int main()
{
    int v = 10;
    printf("%d",v);
}
```

5. ตัวแปร Global ใช้ได้หมด

```
#include<stdio.h>
int v = 10;
int main()
{
    printf("%d",v);
}
```

6. เงื่อนไข

```
#include<stdio.h>
int main()
{
    int j=25;
    if(j>10 && j <= 20)          { printf("1\n"); }
    else if(j>20)                { printf("2\n"); }
    else                          { printf("3\n"); }
}
```

7. วนซ้ำ

```
#include<stdio.h>
int main()
{
    int i;
    for(i=0;i<10;i++)
    {
        printf("%d\n",i);
    }
    return 0;
}
```

8. Comment

```
/* message */
//message
```

9. Array 1 dimension

```
#include<stdio.h>
int main()
{
    int a[5] = { 10, 20, 30, 11, 54 };
    int b[5];
    a[2] = 44;
    printf("%d\n", a[2]);
    return 0;
}
```

| Index | 0 | 1 | 2 | 3 | 4 |
|-------|---|---|---|---|---|
| Value | | | | | |

10. Array 2 dimension

```
#include<stdio.h>
int main()
{
    int b[2][2] = { {16, 2}, {77, 40} };
    int b[2][2];
    b[1][0] = 55;
    printf("%d\n", b[1][0]);
    return 0;
}
```

| Index | 0 | 1 |
|-------|---|---|
| 0 | | |
| 1 | | |

11 խնդ Array

```
int arr[] = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};
int size = sizeof(arr) / sizeof(arr[0]);
```

12. Write File

```
#include<stdio.h>
int main()
{
    int num;
    FILE *f = fopen("d:\\a.txt", "w");
    scanf("%d", &num);
    fprintf(f, "%d", num);
    fclose(f);

    return 0;
}
```

13. Read File

```
#include<stdio.h>
int main()
{
    int num;
    FILE *fptr = fopen("d:\\a.txt","r");
    fscanf(fptr,"%d", &num);
    printf("%d", num);
    fclose(fptr);

    return 0;
}
```

14. String

```
#include<stdio.h>
int main()
{
    char str[] = "Geeks";
    printf("%s\n", str);
    strcpy(str,"change");
    printf("%s\n", str);
    return 0;
}
```

15. แปลง STRING, INT, FLOAT

```
#include<stdio.h>
int main()
{
    int i = atoi("20");
    printf("%d\n",i);

    char buffer [50];
    itoa(250, buffer,10);
    printf("%s\n",buffer);
    return 0;
}
```

16. Pointer

| | | | |
|----------------|------|---------|------|
| name | ip | name | var |
| value | &var | address | &var |
| value of value | *ip | value | 20 |
| address | &ip | | |

| | | | |
|----------------|----------|---------|------|
| name | ip | name | var |
| value | &var | address | &var |
| value of value | *ip = 30 | value | 30 |
| address | &ip | | |

| | | | |
|----------------|------|---------|------|
| name | ip | name | var |
| value | &var | address | &var |
| value of value | *ip | value | 50 |
| address | &ip | | |

```
#include<stdio.h>
int main()
{
    int *ip;
    int var = 20;
    printf("%d\n",var);

    ip = &var;
    printf("%d\n",*ip);
    printf("%d\n",var);
    printf("%x\n",ip);
    printf("%x\n",&var);
    printf("%x\n",&ip);

    *ip = 30;
    printf("%d\n",*ip);
    printf("%d\n",var);

    var = 50;
    printf("%d\n",*ip);
    printf("%d\n",var);

    return 0;
}
```

17. Function

```
#include<stdio.h>
void f1()
{
    printf("me\n");
}
int f2()
{
    return 10;
}
int f3(float a, float b)
{
    float c = a * b;
    return c;
}
int main()
{
    f1();
    printf("%d\n",f2());
    printf("%d\n",f3(5,4));
    return 0;
}
```

18. Pass by Reference and Pass by Value

```
#include<stdio.h>
void f4(int *a)
{
    *a = 50;
    printf("%d\n",*a);
}
void f5(int a)
{
    a = 200;
    printf("%d\n",a);
}
int main()
{
    int b = 44;
    printf("%d\n",b);
    f4(&b);
    printf("%d\n",b);

    b = 44;
    printf("%d\n",b);
    f5(b);
    printf("%d\n",b);
    return 0;
}
```

19. STRUCT

```
#include<stdio.h>
typedef struct B
{
    int i;
    char m[10];
} tB;
void print(tB o)
{
    printf("%d %s \n",o.i,o.m);
}
int main()
{
    tB b1, *b5;

    b1.i = 10;
    strcpy(b1.m,"test");
    print(b1);

    b5 = &b1;
    (*b5).i = 10;
    strcpy( b5->m, "test111");
    print(b1);
    print( (*b5) );
}
```