

:

[]

1. 1-func_ex01.c

•

<pre>#include <stdio.h> int minOf(int x, int y) { return (x < y) ? x : y; } int main() { int a,b; printf(" 2 : "); scanf("%d %d", &a, &b); printf("minOf(%d, %d) = %d\n", a, b, minOf(a, b)); return 0; }</pre>	<pre>2 : 3 2 minOf(3, 2) = 2</pre>

2. 2-func_ex02.c

•

<pre>#include <stdio.h> int sum(int x, int y){ return x + y; } int diff(int x, int y){ return x - y; } int main() { int a,b; printf(" 2 : ");</pre>	<pre>2 : 6 1 sum(6, 1) = 7 diff(6, 1) = 5</pre>

: :

<pre>scanf("%d %d", &a, &b); printf("sum(%d, %d) = %d\n", a, b, sum(a, b)); printf("diff(%d, %d) = %d\n", a, b, diff(a, b)); return 0; }</pre>	
---	--

3. 3-func_ex03.c

•

<pre>#include <stdio.h> int cube(int x){ return x * x * x; } double fcube(double x){ return x * x * x; } int main() { int a; double b; printf(" : "); scanf("%d", &a); printf(" : "); scanf("%lf", &b); printf("cube(%d) = %d\n", a, cube(a)); printf("fcube(%lf) = %lf\n", b, fcube(b)); return 0; }</pre>	<pre>: 3 : 4.3 cube(3) = 27 fcube(4.300000) = 79.507000</pre>

4. 4-func_ex04.c

•

<pre>#include <stdio.h></pre>	<pre>: C</pre>

: :

<pre>void put_nchar(char ch, int no){ for(int i = 0; i < no; i++){ printf("%c", ch); } } int main() { char ch; int no; printf(" : "); scanf(" %c", &ch); printf(" : "); scanf("%d", &no); put_nchar(ch, no); printf("\n"); return 0; }</pre>	<pre> : 5 CCCCC</pre>
--	----------------------------

5. 5-func_ex05.c

•

<pre>#include <stdio.h> void starTriangle(int no){ for(int i = 0; i < no; i++){ for(int j = 0; j <= i; j++){ printf("#"); } printf("\n"); } } int main() { int no; printf(" : "); scanf("%d", &no); starTriangle(no); printf("\n"); return 0; }</pre>	

: :

6. 6-func_ex06.c

•

<pre>#include <stdio.h> int sum(int n) { int total = 0; for (int i = 1; i <= n; i++) { total += i; } return total; } int main() { int n; printf(" : "); scanf("%d", &n); printf("1~%d : %d\n", n, sum(n)); return 0; }</pre>	<pre> : 100 1~100 : 5050</pre>

7. 7-func_ex07.c

•

<pre>#include <stdio.h> void printSquare(int n) { for (int i = 1; i <= n; i++) { for (int j = 1; j <= n; j++) { printf("%d ", i * j); } printf("\n"); } }</pre>	<pre> : 3 1 2 3 2 4 6 3 6 9</pre>

: :

<pre>} int main() { int n; printf(" : "); scanf("%d", &n); printSquare(n); return 0; }</pre>	
---	--

8. 8-func_ex08.c

-

<pre>#include <stdio.h> int power(int base, int exp) { int result = 1; for (int i = 0; i < exp; i++) { result *= base; } return result; } int main() { int base, exp; printf(" : "); scanf("%d", &base); printf(" : "); scanf("%d", &exp); printf("%d\n", power(base, exp)); return 0; }</pre>	<pre>: 2 : 10 1024</pre>

: :

9. 9-func_ex09.c

-

<pre>#include <stdio.h> void swap(int a, int b) { printf("%d %d\n", b, a); } int main() { int a, b; printf(" : "); scanf("%d %d", &a, &b); swap(a, b); return 0; }</pre>	<pre>5 3 : 3 5</pre>

[(Local Variable) (Global Variable)]

	(Local Variable)	(Global Variable)
	({})	
		()
	/ ~	~
		BSS
		0

※

※ [] static

static

#include <stdio.h>

```
void counter() {
    static int count = 0;
```

```

:
:

count++;
printf("count: %d\n", count);
}

int main() {
    counter(); // count: 1
    counter(); // count: 2
    counter(); // count: 3
    return 0;
}

```

[call by value call by reference]

	call by value	call by reference
	void swap(int x, int y) { int tmp = x; x = y; y = tmp; }	void swap(int *px, int *py) { int tmp = *px; *px = *py; *py = tmp; }
	swap(0,1)	swap(px, py)
	X	O

※

: :

[]

1. array_f01.c

-

<pre>#include <stdio.h> void printArr(int arr[], int size) { for (int i = 0; i < size; i++) { printf("a[%d]=%d\n", i, arr[i]); } } int main() { int a[] = {10, 20, 30, 40, 50}; int size = sizeof(a) / sizeof(a[0]); printArr(a, size); return 0; }</pre>	<pre>a[0]=10 a[1]=20 a[2]=30 a[3]=40 a[4]=50</pre>

2. array_f02.c

-

<pre>#include <stdio.h> void reversePrintArr(int arr[], int size) { for (int i = size - 1; i >= 0; i--) { printf("a[%d]=%d\n", i, arr[i]); } } int main() { int a[] = {10, 20, 30, 40, 50}; int size = sizeof(a) / sizeof(a[0]); reversePrintArr(a, size); return 0; }</pre>	<pre>a[4]=50 a[3]=40 a[2]=30 a[1]=20 a[0]=10</pre>

: :

--	--

3. array_f03.c

-

<pre>#include <stdio.h> int minOfArr(int arr[], int size) { int min = arr[0]; for (int i = 1; i < size; i++) { if (arr[i] < min) { min = arr[i]; } } return min; } int main() { int a[] = {33, 67, 23, 87, 95, 47, 75}; int size = sizeof(a) / sizeof(a[0]); printf("min = %d\n", minOfArr(a, size)); return 0; }</pre>	<pre>min = 23</pre>

4. array_f04.c

-

<pre>#include <stdio.h> #include "printArr.h" void arrCpy(double dst[], double src[], int size) { for (int i = 0; i < size; i++) { dst[i] = src[i]; } }</pre>	<pre>arrA : 1.1 2.2 3.3 4.4 5.5 arrB : 1.1 2.2 3.3 4.4 5.5</pre>

: :

```
int main() {
    double arrA[] = {1.1, 2.2, 3.3, 4.4, 5.5};
    double arrB[5];
    int size = sizeof(arrA) / sizeof(arrA[0]);

    printf("arrA : ");
    printArr(arrA, size);

    arrCpy(arrB, arrA, size);

    printf("  \n");

    printf("arrB : ");
    printArr(arrB, size);

    return 0;
}
```

5. array_f05.c

•

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

void shiftLeft(int arr[], int size) {
    int temp = arr[0];
    for (int i = 0; i < size - 1; i++) {
        arr[i] = arr[i + 1];
    }
    arr[size - 1] = temp;
}

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

    printArr(a, size);

    shiftLeft(a, size);
    printf("shiftLeft() 1      \n");
    printArr(a, size);

    shiftLeft(a, size);
    printf("shiftLeft() 2      \n");
    printArr(a, size);
}
```

```
0 1 2 3 4 5 6 7 8 9
shiftLeft() 1
1 2 3 4 5 6 7 8 9 0
shiftLeft() 2
2 3 4 5 6 7 8 9 0 1
```

: :

<pre>return 0; }</pre>	
----------------------------	--