Code Snippets

1.

```
#include<stdio.h>
        void main() {

      int val = 14;
      char *valptr = &val;

      printf("Size : %ld \n", sizeof(*valptr));
          }
```

2.

```
#include<stdio.h>
        void main() {

      int val = 10;
      int *ptr = &val;

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

3.

```
#include<stdio.h>
        void main() {

      int val = 26;
      int* ptr = &val;

      if(ptr == &*ptr)
            printf("Equals \n");
          }
```

4.

```
void main() {

	int val = 41;
	void* ptr = &val;
	printf("%d \n", *((int*)ptr) );

	    }
```

5.

```
#include<stdio.h>

void main(){

	    int data = 10;
	    int *address = &data;

	    printf("data is %d. and address is %d\n", *address, address);

	    *address = ++*address + 5;

	    printf("data is %d \n",data);

}
```

6.

```
#include<stdio.h>

void main(){


        char *string;

        string = "CAFEBABE";

        char **cpointer=&string;

        printf("%c\n",**cpointer);
        printf("%c\n",*(++*cpointer));
        printf("%c\n",*(++*cpointer));
        printf("%c\n",*(++*cpointer));
        printf("%c\n",*(++*cpointer));
        printf("%c\n",*(++*cpointer));
        printf("%c\n",*(++*cpointer));
        printf("%c\n",*(++*cpointer));
}
```

7.

```
#include<stdio.h>
void main(){

        int *ptr1 = 0;
        int *ptr2 = NULL;

        if(ptr1 == ptr2)
                printf("These are Null Pointers\n");
        else
                printf("These are not Null Pointers\n");
```

```
}
```

8.

```
#include<stdio.h>

void main(){

        float a = 12,b = 8.3;
        float *ptr = &a;
        float *ptr1 = &b;

        printf("%f\n",*ptr);
        printf("%f\n",*ptr1);

        a = ptr == ptr1;

        printf("%f\n",*ptr);
        printf("%f\n",*ptr1);
        printf("%f\n",a);

}
```

9.

```
#include<stdio.h>
void main(){

        void a,b;
        void *vptr;

        vptr = &a;

        printf("Pointing to a\n");

        vptr = &b;

        printf("Pointing to b\n");
```

```
}
```

10.

```
#include<stdio.h>
void main(){

        int val = 10;

        const int *ptr = &val;

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

11.

```
#include<stdio.h>
void main() {

    char arr[5] = {'a', 100, 2.3};

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

}
```

12.

```
int printf(const char *, ...);
void main() {

    double arr[] = {22, 67, 43, 619};
    printf("%.0lf \n", *(arr+2));

}
```

13.

```
int printf(const char *, ...);
void main() {

	int *arr[5];

	printf("%ld \n", sizeof(arr));
	printf("%ld \n", sizeof(*arr));
	printf("%ld \n", sizeof(**arr));
}
```

14.

```
#include<stdio.h>
void main() {

	int arr[3] = {10,20,30};

	char *ptr = arr;
	int  *iptr = arr;

	for(int i=0; i<3; i++) {

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

	printf("\n");

	for(int i=0; i<3; i++) {

		printf("%d\n",*(iptr+i));
	}
}
```

15.

```
#include<stdio.h>

void main(){

	int arr[]= {1,2,3};
	arr[0,1,2] = 10;

	for(int i = 0;i < 3;i++)
		printf("%d\n",arr[i]);
}
```

16.

```
int printf(const char *, ...);
void main() {

		int (*ptr) [4];

		printf("%ld \n", sizeof(ptr));
		printf("%ld \n", sizeof(*ptr));

}
```

17.

```
int printf(const char *, ...);
void main() {

		int arr[][] = { {97,98,99,10}, {65,66,67,68} };
		printf("%d \n", arr[1][3]);
```

```
}
```

18.

```
int printf(const char, ...);
void main() {

        char words[][12] = {"swift", "objective-c", "ruby", "go"};
        printf("%c \n", *(*(words+1)+6) );
}
```

19.

```
#include<stdio.h>
void main() {

        const int arr[5] = {1,2,3,4,5};

        int *ptr;

        ptr = arr+3;

        *ptr = 10;

        printf("%d\n",arr[3]);
}
```

20.

```
int printf(const char *, ...);
        void main() {

    float arr[] = {3.14, 4.12, 7.6};
    float *ptr1 = arr + 1;
    float *ptr2 = &arr + 1;

    if(ptr1 == ptr2)
        printf("success! \n");
```

```
}
```

21.

```
#include<stdio.h>
extern int a = 10;
void main() {

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

22.

```
#include<stdio.h>
extern int a;
void main() {
	int a;
	printf("%d \n", a);
}
```

23.

```
// -- file_01.c --
extern int a = 73;

// -- file_02.c --
int printf(const char *, ...);
void main() {

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

```
~:$ gcc file_01.c file_02 && ./a.out
```

24.

```
int printf(const char*, ...);
int a=20;
void main() {

        extern static int a;
        int *ptr = &a;
        a=40;

        printf("%d\n",a);
        printf("%p\n",ptr);
}
```

25.

```
int printf(const char*, ...);
void main() {

        extern int a;
        extern char a;

}
```

26.

```
int printf(const char*, ...);
extern int a;
void main() {

	int i=1;
	int a=4;

	{
		auto int i=2;

		{
			static int i=3;
			printf("%d\n",i);
		}

		printf("%d\n",i);
	}

	printf("%d\n",i);
	printf("%d\n",a);
}
```

27.

```
#include<stdio.h>
void main() {
	void fun(static int a) {
		for(int i=0; i<3; i++) {
			printf("%d\n",a);
			a++;
		}
	}
	fun(2);
}
```

28.

```
#include<stdio.h>
register int a=10;
void main() {

        register int b=15;

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

29.

```
#include<stdio.h>
int add(auto int a, auto int b){

        return a+b;
}
void main(){

        int a = 10,b = 20;

        printf("Addition :- %d\n",add(a,b));

}
```

30.

//file_30a.c

```
#include<stdio.h>
static void fun();
void main(){

        fun();
}
```

// -- file_30b.c

```
#include<stdio.h>
void fun(){

        printf("Function called from Program8a File");
}
```