

C-99 → Returning Pointer from function in C

Declaration for function returning pointer

→ `int * returnPointer (int []);`

(function name)

Returning
integer pointer

accept array
of integer type

Program 1

```
int * returnPointer (int [];
```

```
void main()
```

```
{
```

```
    int *P;
```

```
    int a[] = {1, 2, 3, 4, 5};
```

```
    P = returnPointer(a);
```

```
    printf("v.d", *P);
```

```
}
```

```
int * returnPointer (int a[]).
```

```
{
```

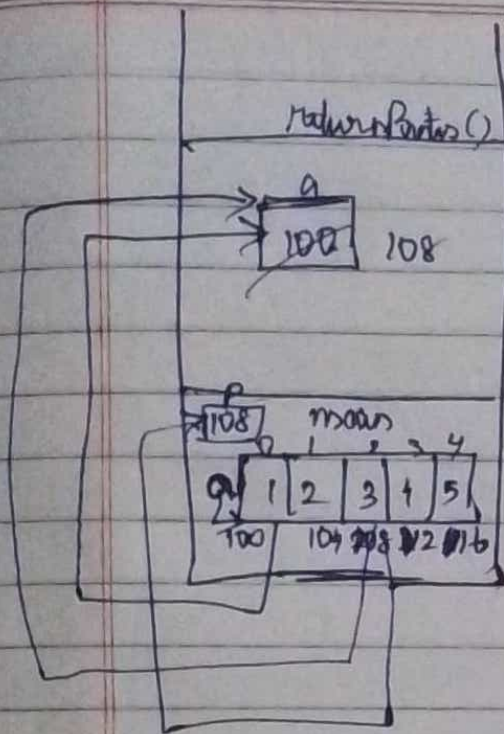
```
    a = a + 2;
```

```
    return a;
```

```
}
```

Q2

3



Program 2.

```
int* returnPointer (int a[]);
```

```
void main()
```

```
{
```

```
int *p;
```

```
int a[] = {1, 2, 3, 4, 5};
```

```
    a = a + 2;
```

```
    printf("%d", a);
```

~~Error~~

Because, name of array is constant pointer to first element of it and we cannot change the pointer reference and it is constant

```
}
int *returnPointer (int a[])
```

```
{
```

```
    a = a + 2;
```

```
    return a;
```

```
}
```

→ No error because 'a' is not array it is only ^{normal} pointer.


```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 1-RETURN POINTER FROM FUNCTION **/
4
5  int* returnpointer(int*);
6
7  int main()
8  {
9      int array[]={1,2,3,4,5};
10     int *arr;
11     int i;
12     arr=returnpointer(array);
13     for(i=0;i<5;i++)
14         printf("Array element in index array[%d] is %d\n",i,array[i]);
15 }
16
17 int* returnpointer(int* arr1)
18 {
19     arr1[0]=2;
20     return arr1;
21 }
22

```

"D:\1. C C++\NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 7_JENNYS LECTURE_FUNCTIONS\FUNCTI

```

Array element in index array[0] is 2
Array element in index array[1] is 2
Array element in index array[2] is 3
Array element in index array[3] is 4
Array element in index array[4] is 5

Process returned 0 (0x0)   execution time : 0.045 s
Press any key to continue.

```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 2-RETURN POINTER FROM FUNCTION **/
4
5  int* returnpointer(int*);
6
7  int main()
8  {
9      int array[]={1,2,3,4,5};
10     int *arr;
11     arr=returnpointer(array);
12     printf("In main: Value at address of arr is %d\n",*arr);
13     getch();
14 }
15
16 int* returnpointer(int* arr1)
17 {
18     printf("In called:Value at address of arr1 is %d\n",*arr1);
19     arr1=arr1+2;
20     return arr1;
21 }
22
23

```

"D:\1. C C++\NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 7_JENNYS LECTURE_FUNCTIONS\FUNCTI...

```
In called: Value at address of arr1 is 1
In main: Value at address of arr is 3
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 3-RETURN POINTER FROM FUNCTION **/
4
5  int* returnpointer();
6
7  int main()
8  {
9      int* p;
10     p=returnpointer();
11     printf("value at address of p is %d\n",*p);
12 }
13
14 int* returnpointer()
15 {
16     int a[]={1,2,3,4,5}; //array is local to this function
17     return &a[2];
18 }
19 //so the array is vanished after it comes out of it and hence when we access it through
20 // pointer from main function then we get no output
21
```

"D:\1. C C++\NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 7_JENNYS LECTURE_FUNCTIONS\FUNCTI...

```
Process returned -1073741819 (0xC0000005)   execution time : 0.439 s
Press any key to continue.
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 3-RETURN POINTER FROM FUNCTION **/
4
5  int* returnpointer();
6
7  int main()
8  {
9      int* p;
10     p=returnpointer();
11     printf("value at address of p is %d\n",*p);
12 }
13
14 int* returnpointer()
15 {
16     int a[]={1,2,3,4,5}; //array is local to this function
17     return &a[2];
18 }
19 //so the array is vanished after it comes out of it and hence when we access it through
20 // pointer from main function then we get no output
21
```

"D:\1. C C++\NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 7_JENNYS LECTURE_FUNCTIONS\FUNCTI...

Process returned -1073741819 (0xC0000005) execution time : 0.528 s
Press any key to continue.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 4-RETURN POINTER FROM FUNCTION **/
4
5  int* returnpointer();
6
7  int main()
8  {
9      int* p;
10     p=returnpointer();
11     printf("value at address of p is %d\n",*p);
12 }
13
14 int* returnpointer()
15 {
16     static int a[]={1,2,3,4,5}; //when static then array a is accessible anytime
17     return &a[2];
18 }
19
20
21
```

"D:\1. C C++\NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 7_JENNYS LECTURE_FUNCTIONS\FUNCTI...

value at address of p is 3

Process returned 0 (0x0) execution time : 0.016 s
Press any key to continue.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 5-RETURN POINTER FROM FUNCTION **/
4
5  int* returnpointer(int*);
6
7  int main()
8  {
9      int a[]={1,2,3,4,5};
10     int *p;
11     a=a+2; //here we cannot change, because it is array pointer whose base address is constant
12     p=returnpointer(a);
13     printf("%d\n", *p);
14     getch();
15 }
16
17 int* returnpointer(int* a)
18 {
19     a=a+2; //we can change the pointer variable
20     return a;
21 }

```

logs & others		
Code::Blocks	Search results	Cccc
Build log	Build messages	CppCheck/Vera++
CppCheck/Vera++ messages	Cscop	
File	Line	Message
D:\1. C C++...		In function 'main':
D:\1. C C++...	11	error: assignment to expression with array type
D:\1. C C++...	14	warning: implicit declaration of function 'getch'; did you mean 'getc'? [-W...
=== Build failed: 1 error(s), 1 warning(s) (0 minute(s), 0 second(s)) ===		