

Day : Pointers (9-8-2025)

1. Write a program to print the address of a variable using pointer.

Input: Get a variable

Process: Assign address

Output: Print the variable

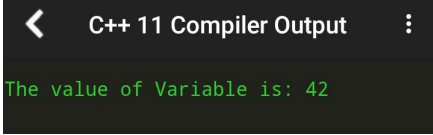
Code:

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

```
int main() {  
    int b = 42;  
    int *a;  
    a = &b;
```

```
    printf("The value of Variable is: %d\n", b);
```

```
    return 0;  
}
```

A screenshot of a terminal window titled "C++ 11 Compiler Output". The output text is "The value of Variable is: 42" displayed in green on a dark background.

```
< C++ 11 Compiler Output  
The value of Variable is: 42
```

2. Write a program to access array elements using pointers.

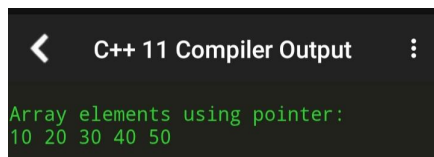
Input: Get an array

Process: Print the address

Output: same array

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

```
int main() {  
    int a[] = {10, 20, 30, 40, 50};  
    int *p = a;  
  
    printf("Array elements using pointer:\n");  
    for (int i = 0; i < 5; i++) {  
        printf("%d ", *(p + i));  
    }  
  
    return 0;  
}
```



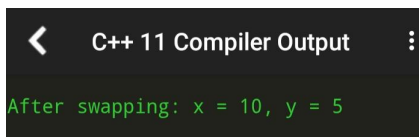
```
< C++ 11 Compiler Output  
Array elements using pointer:  
10 20 30 40 50
```

3. Write a program to swap two numbers using pointers.

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

```
void swap(int *a, int *b) {  
    int temp = *a;  
    *a = *b;  
    *b = temp;  
}
```

```
int main() {  
    int x = 5, y = 10;  
    swap(&x, &y);  
    printf("After swapping: x = %d, y = %d\n", x, y);  
    return 0;  
}
```

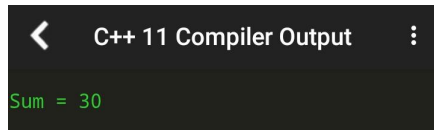


A screenshot of a terminal window titled "C++ 11 Compiler Output". The window has a dark background with a left arrow icon and a vertical ellipsis icon. The output text is "After swapping: x = 10, y = 5" displayed in green.

4. Write a program to add two numbers using pointers.

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

```
int main() {  
    int a = 10, b = 20;  
    int *p1 = &a, *p2 = &b;  
  
    int sum = *p1 + *p2;  
    printf("Sum = %d\n", sum);  
    return 0;  
}
```

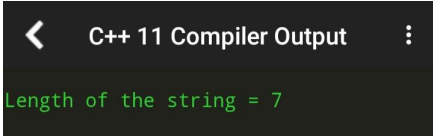
A screenshot of a terminal window titled "C++ 11 Compiler Output". The output shows "Sum = 30" in green text on a black background.

```
< C++ 11 Compiler Output :  
Sum = 30
```

5. Write a program to find the length of a string using pointers.

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

```
int main() {  
    char str[] = "Pointer";  
    char *p = str;  
    int len = 0;  
  
    while (*p != '\0')  
    {  
        len++;  
        p++;  
    }  
  
    printf("Length of the string = %d\n", len);  
    return 0;  
}
```



The screenshot shows a dark-themed window titled "C++ 11 Compiler Output". Inside the window, the text "Length of the string = 7" is displayed in green, indicating the successful execution of the program and the correct calculation of the string length.

6. Write a program to count the vowels using pointer.

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

```
int main() {  
    char str[] = "Hello Pointer World";  
    char *p = str;  
    int vowels = 0;  
  
    while (*p != '\0') {  
        char ch = *p;  
        if (ch == 'a' || ch == 'e' || ch == 'i' ||  
            ch == 'o' || ch == 'u' ||  
            ch == 'A' || ch == 'E' || ch == 'I' ||  
            ch == 'O' || ch == 'U') {  
            vowels++;  
        }  
        p++;  
    }  
  
    printf("Number of vowels = %d\n", vowels);  
    return 0;  
}
```



The screenshot shows a dark-themed window titled "C++ 11 Compiler Output". Inside the window, the text "Number of vowels = 6" is displayed in a light green monospace font.

7. Write a program to reverse a word using pointer.

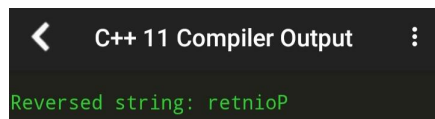
```
#include <stdio.h>
#include <string.h>

int main() {
    char str[] = "Pointer";
    int len = strlen(str);
    char *start = str;
    char *end = str + len - 1;
    char temp;

    while (start < end) {
        temp = *start;
        *start = *end;
        *end = temp;

        start++;
        end--;
    }

    printf("Reversed string: %s\n", str);
    return 0;
}
```

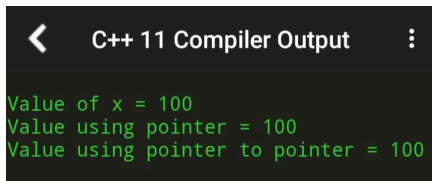


The screenshot shows a dark-themed window titled "C++ 11 Compiler Output". Inside the window, the text "Reversed string: retnioP" is displayed in a green monospaced font.

8. Write a program to demonstrate pointer to pointer.

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

```
int main() {  
    int x = 100;  
    int *ptr = &x;  
    int **pptr = &ptr;  
  
    printf("Value of x = %d\n", x);  
    printf("Value using pointer = %d\n", *ptr);  
    printf("Value using pointer to pointer = %d\n", **pptr);  
    return 0;  
}
```



The screenshot shows the output of a C++ 11 compiler. It displays three lines of text in green on a dark background: "Value of x = 100", "Value using pointer = 100", and "Value using pointer to pointer = 100". The title bar of the window reads "C++ 11 Compiler Output".

```
< C++ 11 Compiler Output  
Value of x = 100  
Value using pointer = 100  
Value using pointer to pointer = 100
```


9. Write a program to allocate memory using malloc() and free it.
10. Write a program to sort an array using pointer notation.