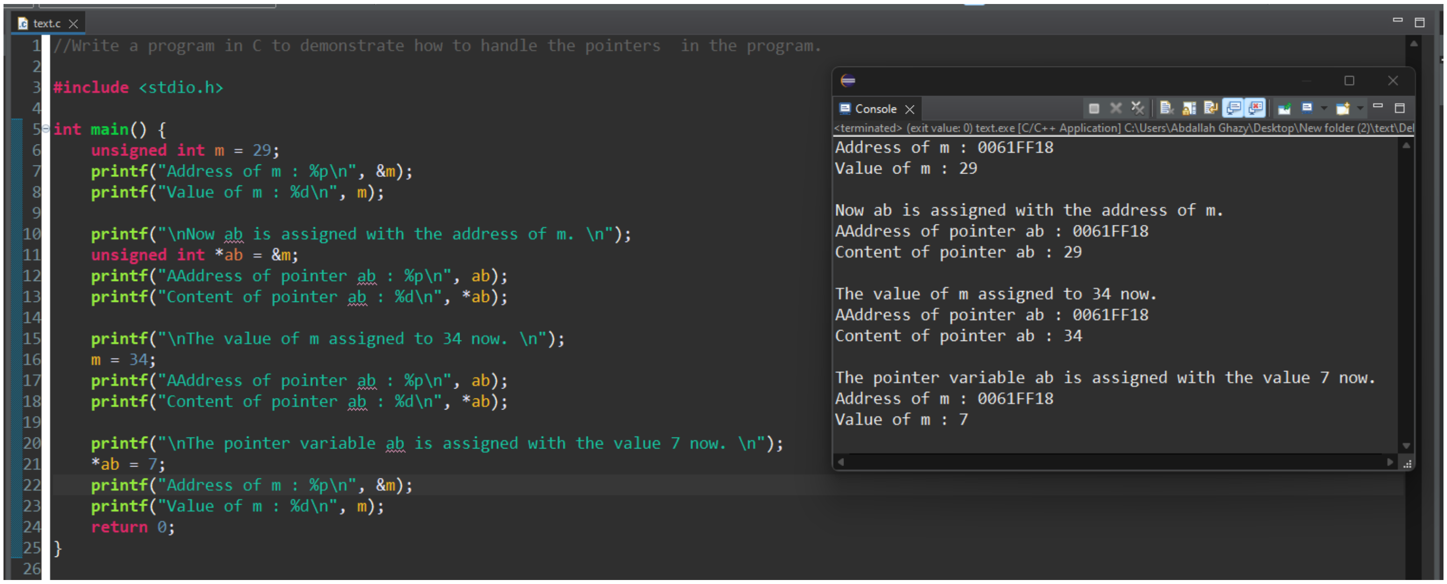
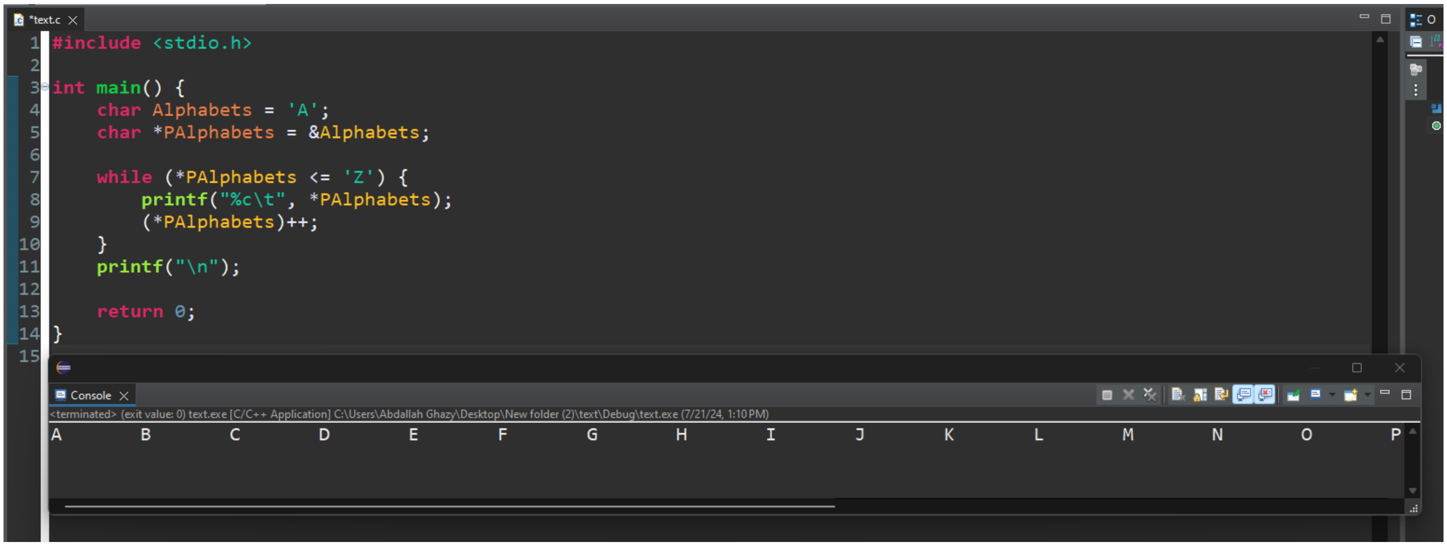
Write a program in C to demonstrate how to handle the pointers in the program.



```c
//Write a program in C to demonstrate how to handle the pointers  in the program.

#include <stdio.h>

int main() {
    unsigned int m = 29;
    printf("Address of m : %p\n", &m);
    printf("Value of m : %d\n", m);

    printf("\nNow ab is assigned with the address of m. \n");
    unsigned int *ab = &m;
    printf("AAddress of pointer ab : %p\n", ab);
    printf("Content of pointer ab : %d\n", *ab);

    printf("\nThe value of m assigned to 34 now. \n");
    m = 34;
    printf("AAddress of pointer ab : %p\n", ab);
    printf("Content of pointer ab : %d\n", *ab);

    printf("\nThe pointer variable ab is assigned with the value 7 now. \n");
    *ab = 7;
    printf("Address of m : %p\n", &m);
    printf("Value of m : %d\n", m);
    return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\Desktop\New folder (2)\text\Del
Address of m : 0061FF18
Value of m : 29

Now ab is assigned with the address of m.
AAddress of pointer ab : 0061FF18
Content of pointer ab : 29

The value of m assigned to 34 now.
AAddress of pointer ab : 0061FF18
Content of pointer ab : 34

The pointer variable ab is assigned with the value 7 now.
Address of m : 0061FF18
Value of m : 7
```

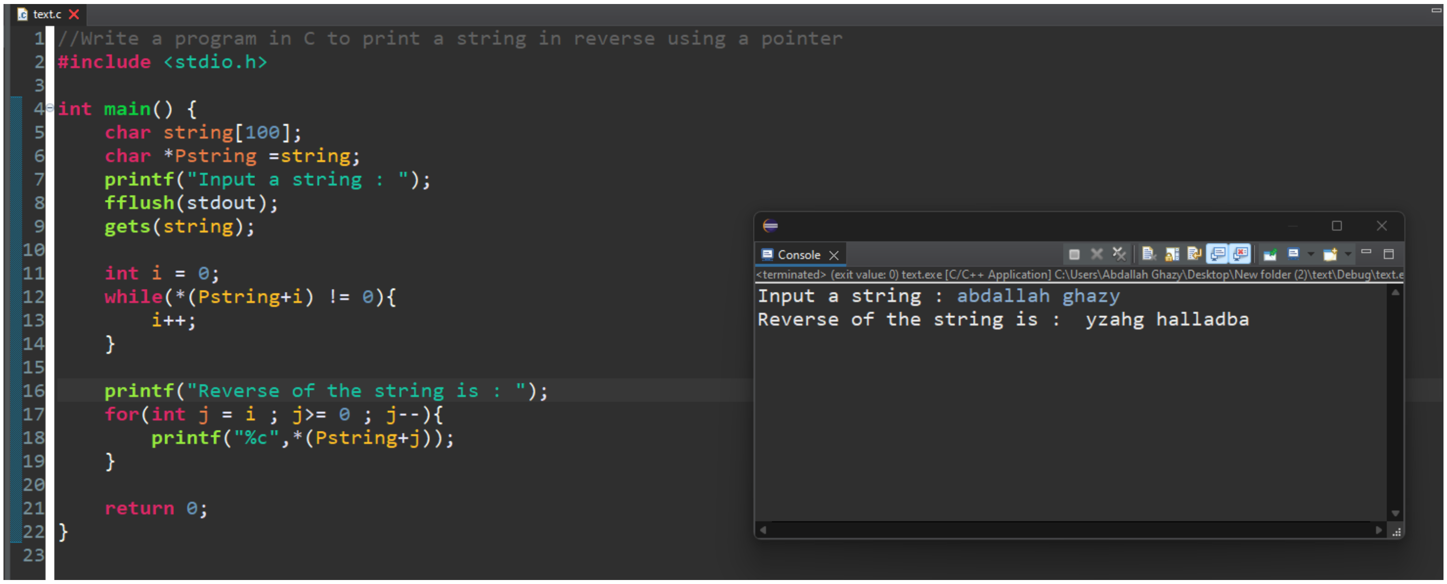
Write a program in C to print all the alphabets using a pointer. Go to the editor



```c
#include <stdio.h>

int main() {
    char Alphabets = 'A';
    char *PAlphabets = &Alphabets;

    while (*PAlphabets <= 'Z') {
        printf("%c\t", *PAlphabets);
        (*PAlphabets)++;
    }
    printf("\n");

    return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\Desktop\New folder (2)\text\Debug\text.exe (7/21/24, 1:10 PM)
A       B       C       D       E       F       G       H       I       J       K       L       M       N       O       P
```

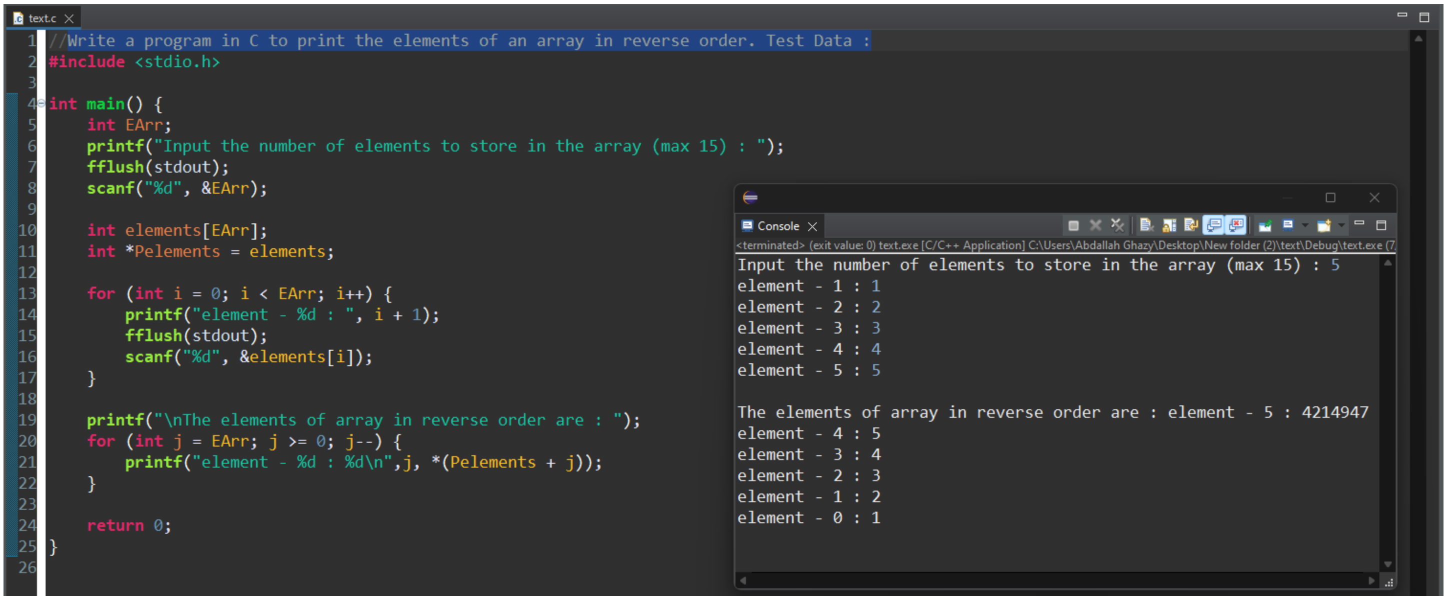
Write a program in C to print a string in reverse using a pointer



```c
//Write a program in C to print a string in reverse using a pointer
#include <stdio.h>

int main() {
    char string[100];
    char *Pstring =string;
    printf("Input a string : ");
    fflush(stdout);
    gets(string);

    int i = 0;
    while(*(Pstring+i) != 0){
        i++;
    }

    printf("Reverse of the string is : ");
    for(int j = i ; j>= 0 ; j--){
        printf("%c",*(Pstring+j));
    }

    return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\Desktop\New folder (2)\text\Debug\text.e
Input a string : abdallah ghazy
Reverse of the string is :  yzahg halladba
```

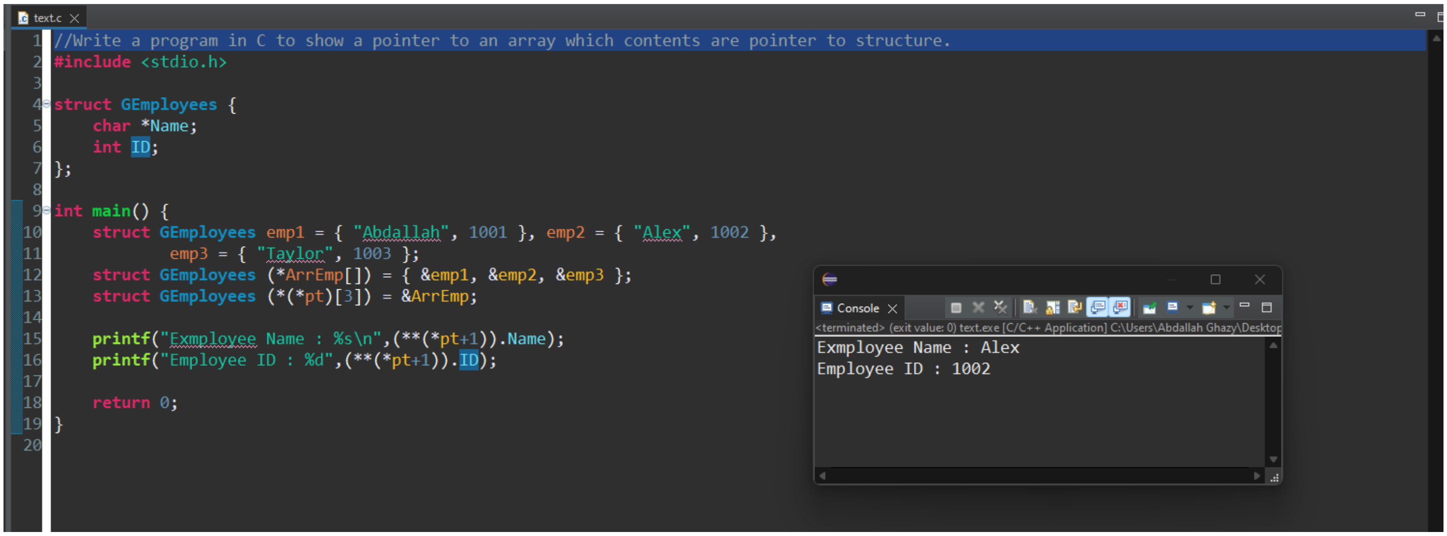
Write a program in C to print the elements of an array in reverse order. Test Data :



```c
//Write a program in C to print the elements of an array in reverse order. Test Data :
#include <stdio.h>

int main() {
    int EArr;
    printf("Input the number of elements to store in the array (max 15) : ");
    fflush(stdout);
    scanf("%d", &EArr);

    int elements[EArr];
    int *Pelements = elements;

    for (int i = 0; i < EArr; i++) {
        printf("element - %d : ", i + 1);
        fflush(stdout);
        scanf("%d", &elements[i]);
    }

    printf("\nThe elements of array in reverse order are : ");
    for (int j = EArr; j >= 0; j--) {
        printf("element - %d : %d\n",j, *(Pelements + j));
    }

    return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\Desktop\New folder (2)\text\Debug\text.exe (7
Input the number of elements to store in the array (max 15) : 5
element - 1 : 1
element - 2 : 2
element - 3 : 3
element - 4 : 4
element - 5 : 5

The elements of array in reverse order are : element - 5 : 4214947
element - 4 : 5
element - 3 : 4
element - 2 : 3
element - 1 : 2
element - 0 : 1
```

Write a program in C to show a pointer to an array which contents are pointer to structure.



```c
//Write a program in C to show a pointer to an array which contents are pointer to structure.
#include <stdio.h>

struct GEmployees {
    char *Name;
    int ID;
};

int main() {
    struct GEmployees emp1 = { "Abdallah", 1001 }, emp2 = { "Alex", 1002 },
            emp3 = { "Taylor", 1003 };
    struct GEmployees (*ArrEmp[]) = { &emp1, &emp2, &emp3 };
    struct GEmployees (*(*pt)[3]) = &ArrEmp;

    printf("Exmployee Name : %s\n",(**(*pt+1)).Name);
    printf("Employee ID : %d",(**(*pt+1)).ID);

    return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\Desktop
Exmployee Name : Alex
Employee ID : 1002
```