

- Wizard.

Write a program in C to take two numbers from the user and print the maximum between two numbers using a pointer.

```
~

#include <stdio.h>
int findMax(int *a, int *b){
    return (*a > *b)? *a:*b;
}
int main(){
    int a,b;
    printf("Enter input: a,b: ");
    scanf("%d,%d", &a,&b);
    printf("The Greater Number Is %d\n", findMax(&a,&b));
}
```

```
~

[wizard@archlinux workshop]$ gcc main.c && ./a.out
Enter input: a,b: 2,3
The Greater Number Is 3
[wizard@archlinux workshop]$
```

Write a program to count and find the sum of all the numbers in the array which are exactly divisible by 7 and not divisible by 2 and 3.

```
~

#include <stdio.h>
int main(){
    int a[] = {7, 14, 21, 28, 35, 42, 49, 63, 70, 77};
    int sum = 0;
    for (int i = 0; i < (sizeof(a)/sizeof(a[0])); i++){
        if (a[i] % 7 == 0 && a[i] %2 != 0 && a[i] % 3 != 0) {
            sum += a[i];
        }
    }
    printf("%d\n", sum);
}
```

```
~

[wizard@archlinux workshop]$ ./a.out
168
[wizard@archlinux workshop]$
```

Write a program to initialize an integer array with values {10,5,7,91,54,24}. Pass this array to a function. If element is found, print out its index number and if element is not found then display the message “Element Not found” in the function.

~

```
#include <stdio.h>

int find(int *arr, int Size, int toFind){
    for (int i = 0; i < Size; i++) {
        if (arr[i] == toFind){
            return i;
        }
    }
    printf("Element Not Found ");
    return -1;
}

int main(){
    int a[] = {10,5,7,91,54,24};
    printf("%d\n", find(a, (sizeof(a)/sizeof(a[0])), 5));
    printf("%d\n", find(a, (sizeof(a)/sizeof(a[0])), 69));
}
```

~

```
[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
1
Element Not Found -1
[wizard@archlinux workshop]$
```

**Write a program to read three integers using an array and find the smallest and largest among them. You must write a function for finding the largest and smallest integer and the result must be displayed in the main function itself. You are not allowed to create global variables.**

---

~

```
#include <stdio.h>

int findSmallest(int *a){
    if (a[0] < a[1] && a[0] < a[2]) {
        return a[0];
    } else if (a[1] < a[2] && a[1] < a[0]) {
        return a[1];
    } else{
        return a[2];
    }
}

int findLargest(int *a){
    if (a[0] > a[1] && a[0] > a[2]) {
        return a[0];
    } else if (a[1] > a[2] && a[1] > a[0]) {
        return a[1];
    } else{
        return a[2];
    }
}

int main(){
    int a[3];
    printf("Enter 3 integers: ");
    scanf("%d, %d, %d",a, a+1, a+2);
    printf("%d\n", findLargest(a));
    printf("%d\n", findSmallest(a));
}
```

~

```
[wizard@archlinux workshop]$ ./a.out
Enter 3 integers: 1,3,5
5
1
[wizard@archlinux workshop]$
```

**Write a program to take three numbers from the user and save it in three different variables. You must swap the value of three numbers using function. You must use call by references.**

---

~

```
#include <stdio.h>

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

int main() {
    int x, y, z;
    printf("Enter three numbers: ");
    scanf("%d %d %d", &x, &y, &z);

    swap(&x, &y, &z);

    printf("After swapping: %d %d %d\n", x, y, z);
    return 0;
}
```

~

```
[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
Enter three numbers: 1 2 3
After swapping: 2 3 1
[wizard@archlinux workshop]$
```

---

**Write a program in C to take three numbers from the user and print the sum, multiplication, quotient, subtraction and minimum between three numbers using a pointer.**

---

```
~

#include <stdio.h>

void sum(int *a, int *b, int *c) {
    printf("Sum :%d\n", *a + *b + *c);
}

void multiply(int *a, int *b, int *c) {
    printf("product :%d\n", *a * *b * *c);
}

void divide(int *a, int *b, int *c) {
    printf("quotent :%d\n", (*a / *b / *c));
}

void subtraction(int *a, int *b, int *c) {
    printf("difference: %d\n", (*a - *b - *c));
}

int main() {
    int x, y, z;
    printf("Enter three numbers: ");
    scanf("%d %d %d", &x, &y, &z);

    sum(&x, &y, &z);
    multiply(&x, &y, &z);
    divide(&x, &y, &z);
    subtraction(&x, &y, &z);
    return 0;
}
```

```
~

[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
Enter three numbers: 1 2 3
Sum :6
product :6
quotent :0
difference: -4
[wizard@archlinux workshop]$
```

**Write a program to count and find the sum of all the numbers in the array which are exactly divisible by 5 and not divisible by 2 and 3.**

```
~

#include <stdio.h>
int main(){
    int a[] = {7, 14, 21, 28, 35, 42, 49, 63, 70, 77};
    int sum = 0;
    for (int i = 0; i < (sizeof(a)/sizeof(a[0])); i++){
        if (a[i] % 5 == 0 && a[i] %2 != 0 && a[i] % 3 != 0) {
            sum += a[i];
        }
    }
    printf("%d\n", sum);
}
```

```
~

[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
35
[wizard@archlinux workshop]$
```

**Write a program to replace each element in an array with the integer value 5.**

```
~

#include <stdio.h>

int main(){
    int a[] = {0,0,0,0,0,0,0};

    for (int i = 0; i < (sizeof(a)/sizeof(a[0])); i++){
        a[i] = 5;
    }

    for (int i = 0; i < (sizeof(a)/sizeof(a[0])); i++){
        printf("%d\n", a[i]);
    }
}
```

```
~

[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
5
5
5
5
5
5
5
5
[wizard@archlinux workshop]$ Z
```

**Write a program to replace each element in an array with the integer value 5. You must take input from users in the main function and save it in an array. You must also take the size of an array from the user. You must create a function that takes two parameters: an array and length of the array. Your function should not return anything and it should not print anything. You must print before and after replacing an element in an array in the main function.**

~

```
#include <stdio.h>

void replace(int *a, int size) {
    for (int i = 0; i < size; i++){
        a[i] = 5;
    }
}

int main(){
    int counter = 0;
    printf("Enter size: ");
    scanf("%d", &counter);
    int a[counter];

    for (int i = 0; i < counter; i++) {
        printf("Enter number: ", &a[i]);
        scanf("%d", &a[i]);
    }
    for (int i = 0; i < (sizeof(a)/sizeof(a[0])); i++){
        printf("%d\n", a[i]);
    }

    replace(a, (sizeof(a)/sizeof(a[0])));

    for (int i = 0; i < (sizeof(a)/sizeof(a[0])); i++){
        printf("%d\n", a[i]);
    }
}
```

~

```
[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
Enter size: 4
Enter number: 1
Enter number: 2
Enter number: 3
Enter number: 4
1
2
3
4
5
5
5
5
[wizard@archlinux workshop]$
```

**Add a new function in the program you have created in Qno. 3 which takes three parameters: an array, length of the array and value to replace each element in an array. You must receive an integer value in the main function. Your function should not return anything and it should not print anything. You must print a new array in the main function.**

---

~

```
#include <stdio.h>

void replace(int *a, int size, int replace) {
    for (int i = 0; i < size; i++){
        a[i] = replace;
    }
}

int main(){
    int counter = 0;
    printf("Enter size: ");
    scanf("%d", &counter);
    int a[counter];

    for (int i = 0; i < counter; i++) {
        printf("Enter number: ", &a[i]);
        scanf("%d", &a[i]);
    }
    for (int i = 0; i < (sizeof(a)/sizeof(a[0])); i++){
        printf("%d\n", a[i]);
    }

    replace(a, (sizeof(a)/sizeof(a[0])), 69);

    for (int i = 0; i < (sizeof(a)/sizeof(a[0])); i++){
        printf("%d\n", a[i]);
    }
}
```

~

```
[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
Enter size: 4
Enter number: 1
Enter number: 2
Enter number: 3
Enter number: 4
1
2
3
4
69
69
69
69
[wizard@archlinux workshop]$
```

**Write a program in C to take two numbers from the user and print the minimum between two numbers using a pointer.**

---



```
~

#include <stdio.h>

int findMin(int *a, int *b){
    return (*a < *b)? *a:*b;
}

int main(){
    int a,b;
    printf("Enter input: a,b: ");
    scanf("%d,%d", &a,&b);
    printf("The Miniumum Is %d\n", findMin(&a,&b));
}
```

```
~

[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
Enter input: a,b: 3,2
The Miniumum Is 2
[wizard@archlinux workshop]$
```

```
~

[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
Enter input: a,b: 3,2
The Miniumum Is 2
[wizard@archlinux workshop]$
```

**2. Write a program to count and find the sum of all the numbers in the array, which are exactly divisible by 5 and not divisible by 2 and 3. Assign an array to a pointer. Print out address of pointer, array and first element of an array using “%p” formatter.**

```
~

#include <stdio.h>
int main(){
    int a[] = {7, 14, 21, 28, 35, 42, 49, 63, 70, 77};
    int sum = 0;
    for (int i = 0; i < (sizeof(a)/sizeof(a[0])); i++){
        if (a[i] % 5 == 0 && a[i] %2 != 0 && a[i] % 3 != 0) {
            sum += a[i];
        }
    }
    int *p = a;
    printf("Adress of pointer %p\n",&p);
    printf("Adress of array %p\n", p);
    printf(" Address First Element of array %p\n", p+1);

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

```
~

[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
Adress of pointer 0x7ffc739e3ec8
Adress of array 0x7ffc739e3ed0
Address First Element of array 0x7ffc739e3ed4
35
[wizard@archlinux workshop]$
```

**3. Write a program to initialize an integer array with values {20, 15, 87, 71, 24, 34}. Pass this array to a function. Take input from a user to search a number from an array. If element is found, print out its index number and if element is not found then display the message “Element Not found” in the function.**

~

```
#include <stdio.h>

int find(int *arr, int Size, int toFind){
    for (int i = 0; i < Size; i++) {
        if (arr[i] == toFind){
            return i;
        }
    }
    printf("Element Not Found ");
    return -1;
}

int main(){
    int a[] = {20, 15, 87, 71, 24, 34};
    int toFind;
    printf("Enter thing to search for: ");
    scanf("%d",&toFind);
    printf("%d\n", find(a, (sizeof(a)/sizeof(a[0])), toFind));
}
```

~

```
[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
Enter thing to search for: 5
Element Not Found -1
[wizard@archlinux workshop]$ ./a.out
Enter thing to search for: 20
0
[wizard@archlinux workshop]$
```