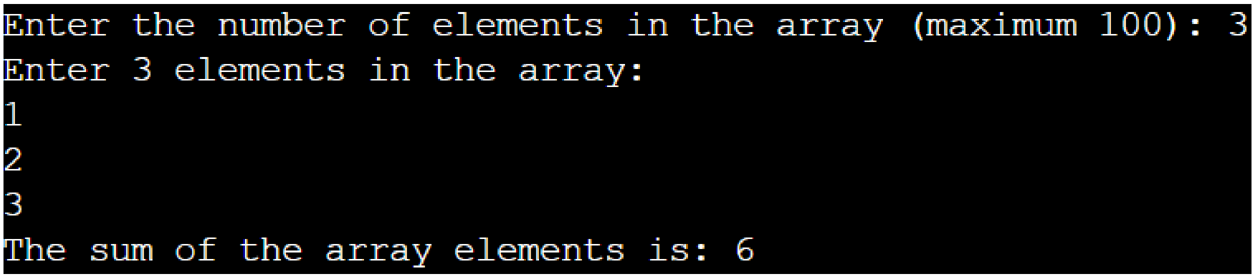
1. Write a C program to find the sum of elements in an array using pointers.

```
#include <stdio.h>

int main() {
    int arr[100];
    int n, i, sum = 0;
    int *ptr;

    printf("Enter the number of elements in the array (maximum 100): ");
    scanf("%d", &n);

    printf("Enter %d elements in the array:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    ptr = arr;


    for (i = 0; i < n; i++) {
        sum += *ptr;
        ptr++;
    }

    printf("The sum of the array elements is: %d\n", sum);

    return 0;
}
```

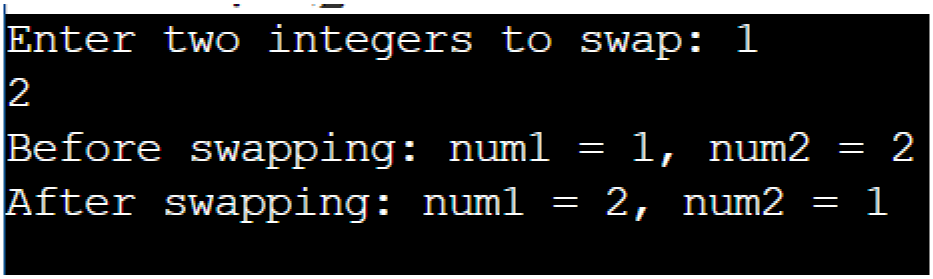


```
Enter the number of elements in the array (maximum 100): 3
Enter 3 elements in the array:
1
2
3
The sum of the array elements is: 6
```

2. Write a C program to swap the values of two integers using pointers.

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

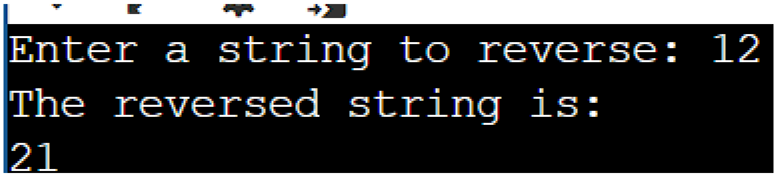
```
void swap(int *x, int *y) {
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}

int main() {
    int num1, num2;

    printf("Enter two integers to swap: ");
    scanf("%d %d", &num1, &num2);

    printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);

    swap(&num1, &num2);

    printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);

    return 0;
}
```



```
Enter two integers to swap: 1
2
Before swapping: num1 = 1, num2 = 2
After swapping: num1 = 2, num2 = 1
```

3. Write a C program to reverse a string using pointers.

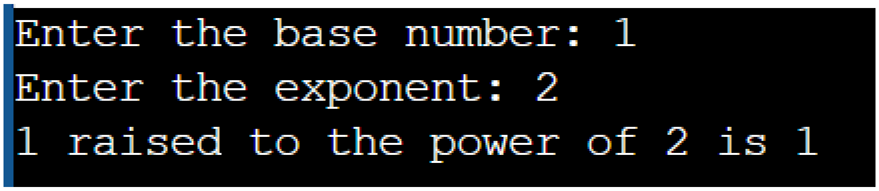
```
#include <stdio.h>

void reverse(char *str) {
  int len = 0, i;
  char *start, *end, temp;


  while (*str != '\0') {
    len++;
    str++;
  }
  start = str - len;
  end = start + len - 1;


  for (i = 0; i < len / 2; i++) {
    temp = *end;
    *end = *start;
    *start = temp;
    start++;
    end--;
  }
}

int main() {
  char str[100];

  printf("Enter a string to reverse: ");
  fgets(str, 100, stdin);

  reverse(str);

  printf("The reversed string is: %s\n", str);

  return 0;
}
```



```
Enter a string to reverse: 12
The reversed string is:
21
```

4. Write a C program to calculate the power of a number using pointers to functions.

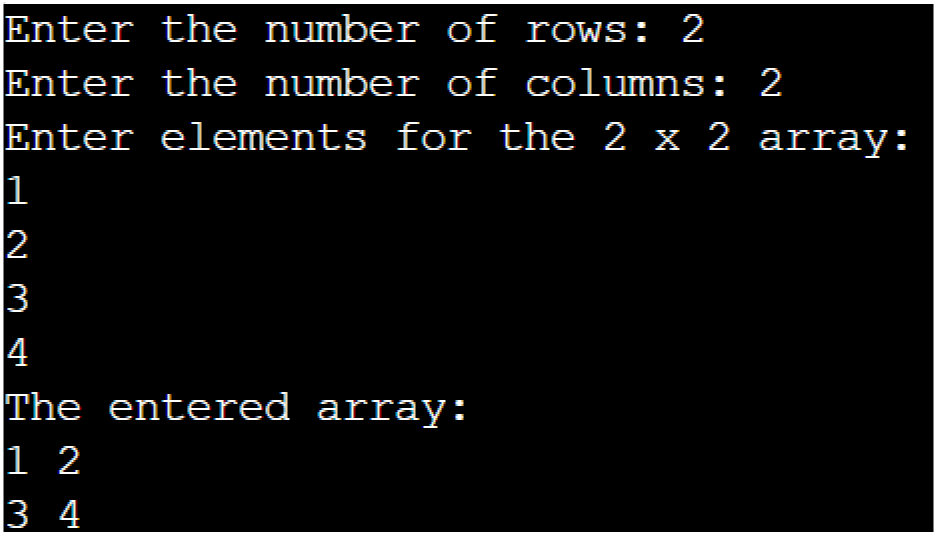
```
#include <stdio.h>

int power(int base, int exp) {
    if (exp == 0) {
        return 1;
    } else {
        return base * power(base, exp - 1);
    }
}

int main() {
    int base, exp, result;

    printf("Enter the base number: ");
    scanf("%d", &base);

    printf("Enter the exponent: ");
    scanf("%d", &exp);


    int (*power_func)(int, int) = power;


    result = power_func(base, exp);

    printf("%d raised to the power of %d is %d\n", base, exp, result);

    return 0;
}
```



```
Enter the base number: 1
Enter the exponent: 2
1 raised to the power of 2 is 1
```

5. Write a C program that dynamically allocates memory for a 2D array based on user input.

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

int main() {
    int rows, cols, i, j;
    int *arr;

    printf("Enter the number of rows: ");
    scanf("%d", &rows);

    printf("Enter the number of columns: ");
    scanf("%d", &cols);
    arr = (int *)malloc(rows * cols * sizeof(int));


    if (arr == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }

    printf("Enter elements for the %d x %d array:\n", rows, cols);
    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            scanf("%d", &arr[i * cols + j]);
        }
    }


    printf("The entered array:\n");
    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            printf("%d ", arr[i * cols + j]);
        }
        printf("\n");
    }


    free(arr);

    return 0;
}
```



```
Enter the number of rows: 2
Enter the number of columns: 2
Enter elements for the 2 x 2 array:
1
2
3
4
The entered array:
1 2
3 4
```