

1. #include <stdio.h>

```
int findSum(int *arr, int size) {
```

```
    int sum = 0;
```

```
    for (int i = 0; i < size; i++) {
```

```
        sum += *(arr + i);
```

```
    }
```

```
    return sum;
```

```
}
```

```
int main() {
```

```
    int arr[] = {7, 2, 9, 8, 4};    int size = sizeof(arr) /
```

```
    sizeof(arr[0]);    int result = findSum(arr, size);    printf("Sum  
of elements in the array: %d\n", result);
```

```
    return 0;
```

```
}
```

The screenshot shows the OnlineGDB web interface. The browser tabs include 'Free Online Document Editing', 'Sign in to your account', and 'Online C Compiler - online editor'. The address bar shows 'onlinegdb.com/online_c_compiler'. The interface has a sidebar on the left with links like 'OnlineGDB beta', 'code, compile, run, debug, share.', 'IDE', 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Jobs', 'Sign Up', 'Login', and 'Learn Python with KodeKloud'. The main area displays the C code from the previous blocks, with line numbers 1 through 26. Below the code editor, the 'Input' field is empty, and the 'Output' section shows the program's execution: 'Sum of elements in the array: 30' followed by '...Program finished with exit code 0' and 'Press ENTER to exit console.'.

```

2. #include <stdio.h>
3.
   void swap(int *p, int *q) {
4.
   int temp = *p;
   *p = *q;
5.   *q = temp;
   }
int main( ) {

int num1 = 67, num2 = 92;
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;
}

```

The screenshot shows the OnlineGDB web interface. The browser tabs include 'Free Online Document Editor', 'Sign in to your account', and 'Online C Compiler - online ed'. The address bar shows 'onlinegdb.com/online_c_compiler'. The interface has a sidebar on the left with links like 'My Projects', 'Classroom', 'Learn Programming', and 'Sign Up'. The main editor area shows the C code from the previous block. Below the editor, the 'Input' section shows the output of the program: 'Before swapping: num1 = 67, num2 = 92' and 'After swapping: num1 = 92, num2 = 67'. At the bottom, it says '...Program finished with exit code 0' and 'Press ENTER to exit console.'.

```

3. #include <stdio.h>
   #include <string.h>
   void reverseString(char *str) {
   int length = strlen(str);
   char *start = str;
   char *end = str + length - 1;
   while (start < end) {
   char temp = *start;
   *start = *end;
   *end = temp;
   start++;

```

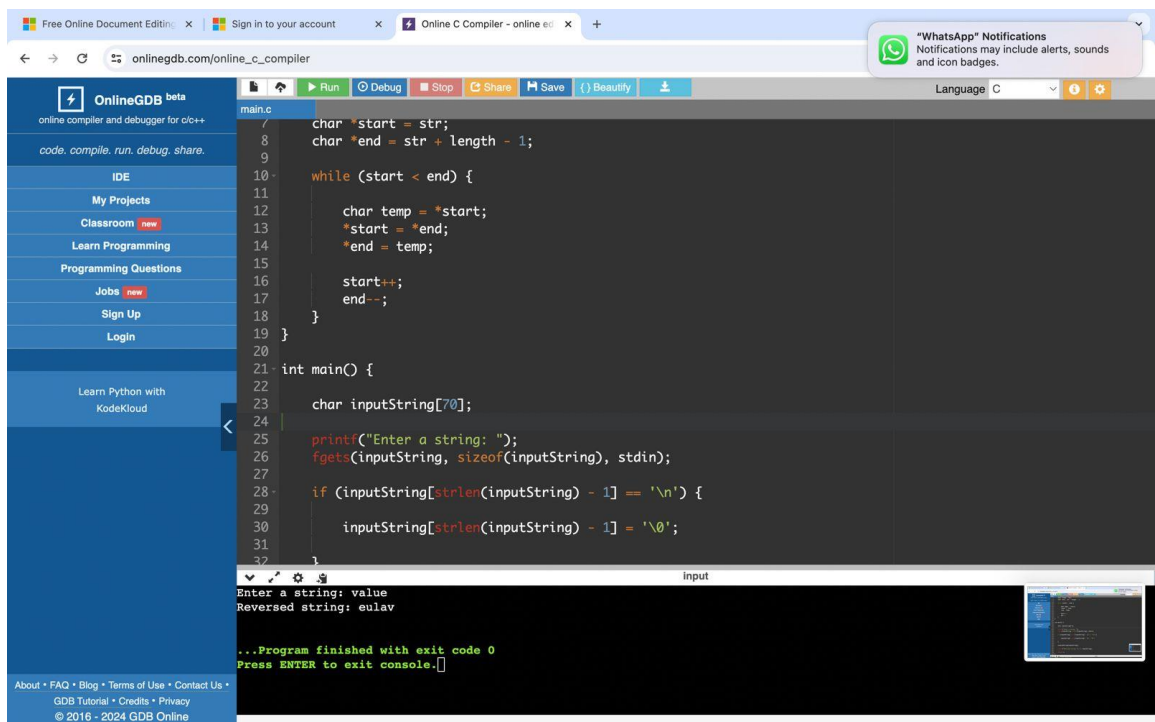
```

end--;
}
}
int min() {

    char inputString[70];
    printf("Enter a string: ");
    fgets(inputString, sizeof(inputString), stdin);
    if (inputString[strlen(inputString) - 1] == '\n') {
        inputString[strlen(inputString) - 1] = '\0';
    }
    reverseString(inputString);
    printf("Reversed string: %s\n", inputString);

    return 0;
}

```



```

4. #include <stdio.h> 5.
    double power(double base, int exponent) {
        double result = 1.0;
        for (int i = 0; i < exponent; i++) {
            result *= base;
        }
        return result; }
typedef double (*PowerFunction)(double, int);
int main() {
    double base, result; 18. int exponent; 1

```

```

printf("Enter base: ");
scanf("%lf", &base);
printf("Enter exponent: ");
scanf("%d", &exponent);
PowerFunction powerPtr = &power;
result = (*powerPtr)(base, exponent);
printf("%.2lf^%d = %.2lf\n", base, exponent, result);
return 0;
}

```

The screenshot shows the OnlineGDB web IDE. The code editor contains the following C program:

```

1 #include <stdio.h>
2
3 double power(double base, int exponent) {
4     double result = 1.0;
5     for (int i = 0; i < exponent; i++) {
6         result *= base;
7     }
8     return result;
9 }
10
11 typedef double (*PowerFunction)(double, int);
12
13 int main() {
14     double base, result;
15     int exponent;
16
17     printf("Enter base: ");
18     scanf("%lf", &base);
19
20     printf("Enter exponent: ");
21     scanf("%d", &exponent);
22
23     PowerFunction powerPtr = &power;
24
25     result = (*powerPtr)(base, exponent);
26
27     printf("%.2lf^%d = %.2lf\n", base, exponent, result);
28     return 0;
29 }

```

The console output shows the program execution:

```

Enter base: 7
Enter exponent: 8
7.00^8 = 5764801.00
...Program finished with exit code 0
Press ENTER to exit console.

```

```

5.    #include<stdio.h>
#include <stdlib.h>
int main( ) {    int
rows, cols; .
    printf("Enter the number of rows: ");
    scanf("%d", &rows);

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

    int **matrix = (int **)malloc(rows * sizeof(int *));
    for (int i = 0; i < rows; i++) {
        matrix[i] = (int *)malloc(cols * sizeof(int));
    }

    printf("Enter elements of the matrix:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {

```

```

scanf("%d", &matrix[i][j]);
}
}28.
printf("Matrix:\n");
for (int i = 0; i < rows; i++) {
for (int j = 0; j < cols; j++) {
printf("%d\t", matrix[i][j]);
}
printf("\n");
}
for (int i = 0; i < rows; i++) {

free(matrix[i]);

}

free(matrix);
return 0;
}

```

The screenshot displays the OnlineGDB web IDE interface. The code editor shows a C program that takes the number of rows and columns as input, dynamically allocates a 2D matrix, fills it with user-provided values, prints the matrix, and then frees the allocated memory. The console output at the bottom shows the program's execution, including the input values (3 rows, 4 columns) and the resulting 3x4 matrix.

```

main.c
10 printf("Enter the number of columns: ");
11 scanf("%d", &cols);
12
13 int **matrix = (int **)malloc(rows * sizeof(int *));
14 for (int i = 0; i < rows; i++) {
15     matrix[i] = (int *)malloc(cols * sizeof(int));
16 }
17
18 printf("Enter elements of the matrix:\n");
19 for (int i = 0; i < rows; i++) {
20     for (int j = 0; j < cols; j++) {
21         scanf("%d", &matrix[i][j]);
22     }
23 }
24
25 printf("Matrix:\n");
26 for (int i = 0; i < rows; i++) {
27     for (int j = 0; j < cols; j++) {
28         printf("%d\t", matrix[i][j]);
29     }
30     printf("\n");
31 }
32
33 for (int i = 0; i < rows; i++) {
34
35     free(matrix[i]);
36 }
37
38 free(matrix);
39 return 0;
40 }

```

Input:

```

3 5 7 9
2 3 4 5
Matrix:
3      5      7      9
2      3      4      5

```

...Program finished with exit code 0
Press ENTER to exit console.