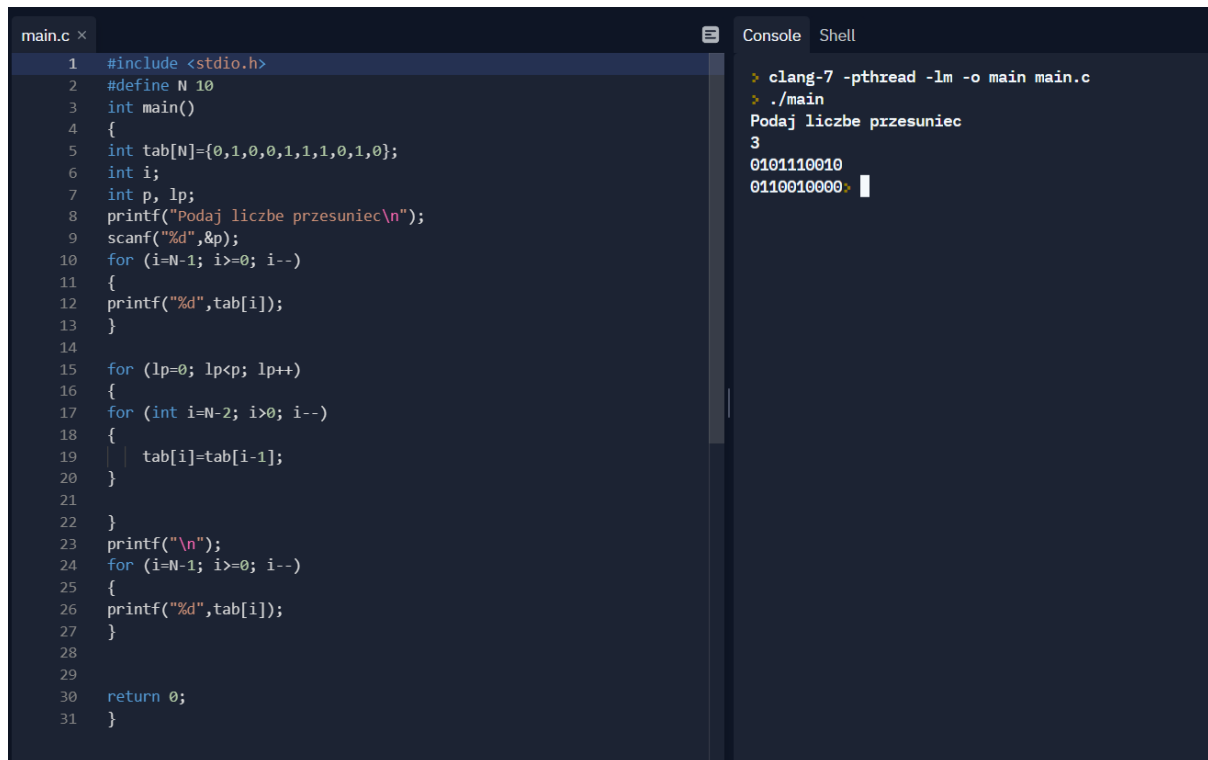


Zadanie 1



The screenshot shows a code editor with a file named 'main.c'. The code is a C program that defines a constant N as 10, initializes an array 'tab' with values {0, 1, 0, 0, 1, 1, 1, 0, 1, 0}, and prompts the user to enter a shift value 'p'. It then prints the array elements in reverse order and shifts them to the right by 'p' positions. The console output shows the program being compiled with 'clang-7 -pthread -lm -o main main.c', the user entering '3', and the resulting array values '0101110010' and '0110010000'.

```
main.c x
1 #include <stdio.h>
2 #define N 10
3 int main()
4 {
5     int tab[N]={0,1,0,0,1,1,1,0,1,0};
6     int i;
7     int p, lp;
8     printf("Podaj liczbe przesuniec\n");
9     scanf("%d",&p);
10    for (i=N-1; i>=0; i--)
11    {
12        printf("%d",tab[i]);
13    }
14
15    for (lp=0; lp<p; lp++)
16    {
17        for (int i=N-2; i>0; i--)
18        {
19            tab[i]=tab[i-1];
20        }
21    }
22    printf("\n");
23    for (i=N-1; i>=0; i--)
24    {
25        printf("%d",tab[i]);
26    }
27
28
29
30    return 0;
31 }
```

Console Shell

```
> clang-7 -pthread -lm -o main main.c
> ./main
Podaj liczbe przesuniec
3
0101110010
0110010000
```

Kod:

```
#include <stdio.h>
#define N 10
int main()
{
    int tab[N]={0,1,0,0,1,1,1,0,1,0};
    int i;
    int p, lp;
    printf("Podaj liczbe przesuniec\n");
    scanf("%d",&p);
    for (i=N-1; i>=0; i--)
    {
        printf("%d",tab[i]);
    }

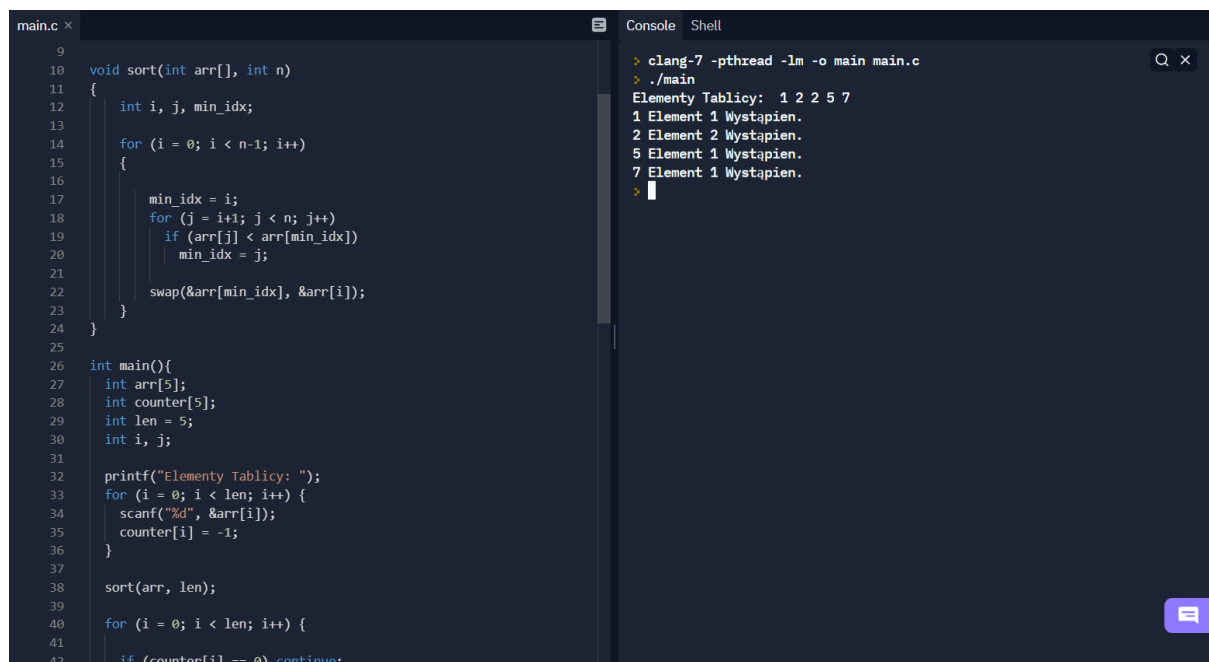
    for (lp=0; lp<p; lp++)
    {
        for (int i=N-2; i>0; i--)
        {
            tab[i]=tab[i-1];
        }
    }

    printf("\n");
```

```
for (i=N-1; i>=0; i--)
{
printf("%d",tab[i]);
}
```

```
return 0;
}
```

Zadanie 2



The screenshot shows a code editor with a file named `main.c`. The code implements a selection sort algorithm. The `sort` function takes an array `arr` and its size `n`, and sorts it in ascending order. The `main` function initializes an array `arr` with values `1, 2, 2, 5, 7`, prints it, sorts it, and prints it again. The console window on the right shows the output of the program, which is the sorted array: `1 2 2 5 7`.

```
main.c x
9
10 void sort(int arr[], int n)
11 {
12     int i, j, min_idx;
13
14     for (i = 0; i < n-1; i++)
15     {
16         min_idx = i;
17         for (j = i+1; j < n; j++)
18             if (arr[j] < arr[min_idx])
19                 min_idx = j;
20         swap(&arr[min_idx], &arr[i]);
21     }
22 }
23
24
25
26 int main(){
27     int arr[5];
28     int counter[5];
29     int len = 5;
30     int i, j;
31
32     printf("Elementy Tablicy: ");
33     for (i = 0; i < len; i++) {
34         scanf("%d", &arr[i]);
35         counter[i] = -1;
36     }
37
38     sort(arr, len);
39
40     for (i = 0; i < len; i++) {
41         if (counter[i] == 0) continue;
42     }
```

```
Console Shell
> clang-7 -pthread -lm -o main main.c
> ./main
Elementy Tablicy: 1 2 2 5 7
1 Element 1 Wystapien.
2 Element 2 Wystapien.
5 Element 1 Wystapien.
7 Element 1 Wystapien.
>
```

Kod:

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

```
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}
```

```
void sort(int arr[], int n)
{
    int i, j, min_idx;

    for (i = 0; i < n-1; i++)
    {
```

```
        min_idx = i;
        for (j = i+1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;

        swap(&arr[min_idx], &arr[i]);
    }
}
```

```
int main(){
    int arr[5];
    int counter[5];
    int len = 5;
    int i, j;

    printf("Elementy Tablicy: ");
    for (i = 0; i < len; i++) {
        scanf("%d", &arr[i]);
        counter[i] = -1;
    }

    sort(arr, len);

    for (i = 0; i < len; i++) {

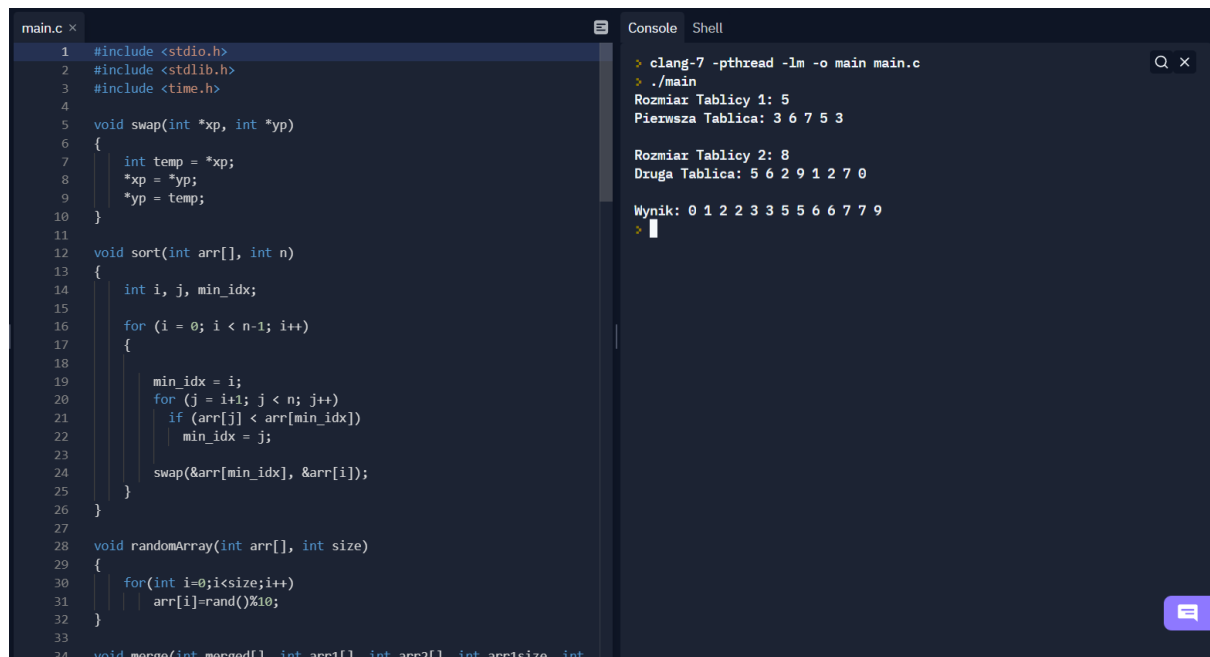
        if (counter[i] == 0) continue;

        counter[i] = 1;
        for (j = i + 1; j < len; j++) {
            if(arr[i] == arr[j]) {
                counter[i]++;
                counter[j] = 0;
            }
        }
    }

    for (i = 0; i < len; i++) {
        if (counter[i] == 0) continue;
        printf("%d Element %d Wystapien.\n", arr[i], counter[i]);
    }

    return 0;
}
```

Zadanie 3



```
main.c x
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 void swap(int *xp, int *yp)
6 {
7     int temp = *xp;
8     *xp = *yp;
9     *yp = temp;
10 }
11
12 void sort(int arr[], int n)
13 {
14     int i, j, min_idx;
15     for (i = 0; i < n-1; i++)
16     {
17         min_idx = i;
18         for (j = i+1; j < n; j++)
19             if (arr[j] < arr[min_idx])
20                 min_idx = j;
21         swap(&arr[min_idx], &arr[i]);
22     }
23 }
24
25 void randomArray(int arr[], int size)
26 {
27     for(int i=0;i<size;i++)
28         arr[i]=rand()%10;
29 }
30
31 void merge(int merged[], int arr1[], int arr2[], int arr1size, int
```

```
> clang-7 -pthread -lm -o main main.c
> ./main
Rozmiar Tablicy 1: 5
Pierwsza Tablica: 3 6 7 5 3

Rozmiar Tablicy 2: 8
Druza Tablica: 5 6 2 9 1 2 7 0

Wynik: 0 1 2 2 3 3 5 5 6 6 7 7 9
>
```

Kod:

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

```
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}
```

```
void sort(int arr[], int n)
{
    int i, j, min_idx;

    for (i = 0; i < n-1; i++)
    {

        min_idx = i;
        for (j = i+1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;

        swap(&arr[min_idx], &arr[i]);
    }
}
```

```
void randomArray(int arr[], int size)
{
    for(int i=0;i<size;i++)
        arr[i]=rand()%10;
}

void merge(int merged[], int arr1[], int arr2[], int arr1size, int arr2size, int arr3size)
{
    for(int i = 0; i < arr1size; i++)
    {
        merged[i] = arr1[i];
    }

    for(int i = 0, j = arr1size; j < arr3size && i < arr2size; i++, j++)
    {
        merged[j] = arr2[i];
    }
}

void printArray(int arr[], int size)
{
    for (int i = 0; i < size; i++)
        printf("%d ", arr[i]);

    printf("\n");
}

int main()
{
    int arr1[2048];
    int arr1len;

    int arr2[2048];
    int arr2len;

    int arr3[2048];

    printf("Rozmiar Tablicy 1: ");
    scanf("%d", &arr1len);

    randomArray(arr1, arr1len);

    printf("Pierwsza Tablica: ");
    printArray(arr1, arr1len);

    printf("\n");

    printf("Rozmiar Tablicy 2: ");
```

```
scanf("%d", &arr2len);

randomArray(arr2, arr2len);

printf("Druga Tablica: ");
printArray(arr2, arr2len);

printf("\n");

int arr3len = (arr1len + arr2len);

merge(arr3, arr1, arr2, arr1len, arr2len, arr3len);

sort(arr3, arr3len);

printf("Wynik: ");

printArray(arr3, arr3len);

return 0;
}
```