

**PWSZ Elbląg**  
**Instytut Informatyki Stosowanej im. Krzysztofa**  
**Brzeskiego**

**Podstawy programowania I – laboratorium**

Studium Stacjonarne, sem. 1, 2021/2022

**Sprawozdanie nr 9,**  
**środa grupa V, 10:45 – 12:15**

**Data wykonania ćwiczenia: 30.01.2022**

**Data oddania sprawozdania: 31.01.2022**

**Nazwisko i imię: Gajewski Mikołaj**

**Nr albumu: 20180**

**Nazwa pliku : lab9\_SD\_Gajewski\_Mikołaj\_20180.pdf**

1. Napisać program przesuwający w lewo o zadaną liczbę pozycji zawartość tablicy  $a[N]$ , której elementy są cyframi binarnymi. Przyjąć, że na pozycje o najniższych wagach wchodzi zera, a wartość pozycji o indeksie  $N-1$  nie powinna się zmieniać (jest to tzw. powielanie znaku liczby w systemach stosujących najwyższy bit jako bit znaku liczby).

Rozwiązanie :

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

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

```
#define N 8
```

```
/* run this program using the console pauser or add your own getch, system("pause") or  
input loop */
```

```
int main(int argc, char *argv[]) {
```

```
    int i,x;
```

```
    int tab[N]={1,1,1,1,1,1,1,1};
```

```
    printf("Wyświetlenie aktualnej tabeli : ");
```

```
    for(i=0; i<N; i++)
```

```
    {
```

```
        printf("\ntab[%d]=%d",i,tab[i]);
```

```
    }
```

```
    for( i=N-1; i>0 ; i--){
```

```
        tab[i]=tab[i-1];
```

```
    }
```

```
    tab[0]=0;
```

```
    printf("\n");
```

```
    for(i=0; i<N; i++)
```

```
    {
```

```

        printf("\ntab[%d]=%d",i,tab[i]);

    }

    printf("\n");

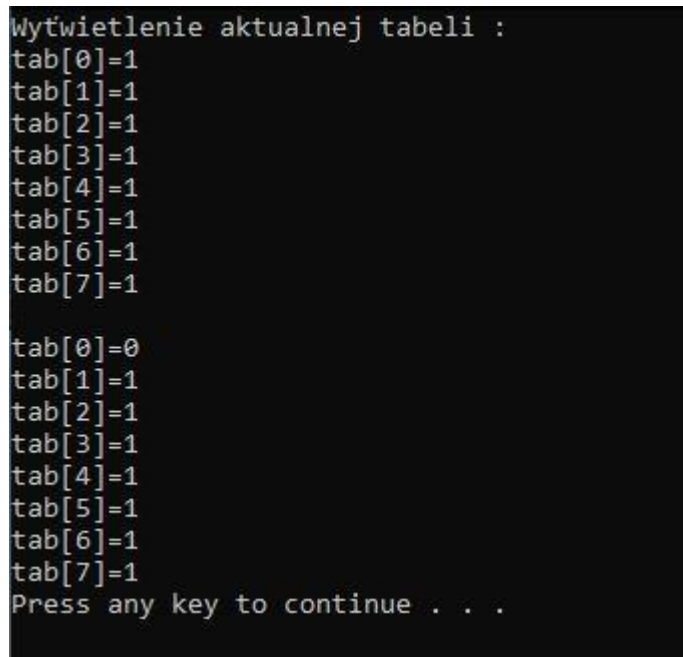
    system("pause");

    return 0;

}

```

Widok z terminala :



```

Wytwietlenie aktualnej tabeli :
tab[0]=1
tab[1]=1
tab[2]=1
tab[3]=1
tab[4]=1
tab[5]=1
tab[6]=1
tab[7]=1

-----
tab[0]=0
tab[1]=1
tab[2]=1
tab[3]=1
tab[4]=1
tab[5]=1
tab[6]=1
tab[7]=1
Press any key to continue . . .

```

2. Wypełnić tablicę liczbami pseudolosowymi o wartościach z przedziału[0,20]i posortować nierosnąco przy zastosowaniu algorytmu sortowania bąbelkowego.

Rozwiązanie :

```

#define N 5 // sortowanie niemalejąco

#include <stdio.h>

#include <conio.h>

int main(int argc, char **argv){

    int i , nr=1,p, zam;

    int n=20;

    int a[N];

```

```

int b[N];

printf("\nPrzed sortowaniem : ");

srand(time(NULL));

for(i=0;i<N;i++)
{
    a[i]=1+rand()%20;

    printf("\ntab[%d]=%d",i,a[i]);
}

printf("\n");

do {
    zam=0;

    for (i=0;i<N-1;i++)
        if (a[i+1]<a[i]){
            p=a[i];
            a[i]=a[i+1];
            a[i+1]=p;
            zam=1;
        }

}while (zam);

printf("\n Stan tablicy po sortowaniu");

for ( i = 0; i < N; i++)
    b[i]=a[N-1-i];

printf("\n");

for ( i =0; i < N ; i++)
    printf("\ntab[%d]=%d",i, b[i]);

getchar();

```

```
}
```

Widok z terminala :

```
Przed sortowaniem :
tab[0]=13
tab[1]=11
tab[2]=11
tab[3]=14
tab[4]=17

Stan tablicy po sortowaniu

tab[0]=17
tab[1]=14
tab[2]=13
tab[3]=11
tab[4]=11
```

3. Wczytać tablicę liczb typu double i posortować nierosnąco przy zastosowaniu algorytmu sortowania przez proste wybieranie.

Rozwiązanie :

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

```
#include <conio.h>
```

```
#define N 8
```

```
int main(int argc, char argv[]){
```

```
    int i,i1,j,k,p,amin,sizeHalf,med;
```

```
    double tab[N]= {9,5,-2,7,-4,15,-20,100};
```

```
    double b[N];
```

```
        for ( i =0; i < N ; i ++){
```

```
            printf("\ntab[%d]=%f",i, tab[i]);
```

```
        }
```

```
        printf("\n");
```

```
        for (i=0;i<N;i++){
```

```
            amin=tab[i];
```

```
            k=i;
```

```

        for (j=i+1;j<N;j++)
        if (tab[j]<amin){
            amin=tab[j];
            k=j; }
        if (k!=i){
            p=tab[i];
            tab[i]=tab[k];
            tab[k]=p;
        }
    }

    printf("\n Stan tablicy po sortowaniu ");

    for (i=0;i<N;i++)

        printf(" % f ",tab[i]);

    //odwracanie tablicy
    for ( i = 0; i < N; i++)

        b[i]=tab[N-1-i];

    printf("\n");

    for ( i =0; i < N ; i ++)

        printf("\ntab[%d]=%f",i, b[i]);

    printf("\n");

    system("pause");

}

```

Widok z terminala :

```

tab[0]=9.000000
tab[1]=5.000000
tab[2]=-2.000000
tab[3]=7.000000
tab[4]=-4.000000
tab[5]=15.000000
tab[6]=-20.000000
tab[7]=100.000000

Stan tablicy po sortowaniu  -20.000000  -4.000000  -2.000000  5.000000  7.000000  9.000000  15.000000  100.000000

tab[0]=100.000000
tab[1]=15.000000
tab[2]=9.000000
tab[3]=7.000000
tab[4]=5.000000
tab[5]=-2.000000
tab[6]=-4.000000
tab[7]=-20.000000
Press any key to continue . . .

```

4. Napisać program wyznaczający w tablicy pozycję drugiego wystąpienia elementu o zadanej wartości.

Rozwiązanie :

```

#define N 8

#include <stdio.h>

#include <conio.h>

int main(int argc, char argv[]){

    int i,j;

    int a[N]= {3,5,5,5,-4,7,-20,100};

    int x;

    for ( i =0; i < N ; i ++ )

        printf("\ntab[%d]=%d",i, a[i]);

    printf("\nPodaj wartosc ktora sie powtarza w tablicy : ");

    while(scanf("%d",&x)!=1)

    {

        printf("\nBlad formatu! Prosze podac jeszcze raz : ");

        fflush(stdin);

    }

    printf("\nWczytano : %d",x);

```

```

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

        if(x==a[i]){

            printf("\nZnaleziona kolejna : %d, o indeksie : %d", x,i);

        }

    }

    return 0;

}

```

Widok z terminala :

```

tab[0]=3
tab[1]=5
tab[2]=5
tab[3]=5
tab[4]=-4
tab[5]=7
tab[6]=-20
tab[7]=100
Podaj wartosc ktora sie powtarza w tablicy : 5

Wczytano : 5
Znaleziona kolejna : 5, o indeksie : 1
Znaleziona kolejna : 5, o indeksie : 2
Znaleziona kolejna : 5, o indeksie : 3
-----
Process exited after 4.792 seconds with return value 0
Press any key to continue . . .

```

5. Napisać program wyznaczający medianę we wczytanej tablicy liczb typu int.

Rozwiązanie :

```

#include <stdio.h>

#include <conio.h>

int main(int argc, char argv[]){

    int i,i1,j,k,p,amin,sizeHalf,med;

    int tab[8]= {9,5,-2,7,-4,15,-20,100};

    int sizeOfTab = sizeof(sizeof tab / sizeof *tab);

```



```

for ( i =0; i < 8 ; i ++){

printf("\ntab[%d]=%d",i, tab[i]);

}

printf("\n");

for (i=0;i<8;i++){

    amin=tab[i];

    k=i;

        for (j=i+1;j<8;j++)

            if (tab[j]<amin){

                amin=tab[j];

                k=j; }

                if (k!=i){

                    p=tab[i];

                    tab[i]=tab[k];

                    tab[k]=p;

                }

    }

}

printf("\n Stan tablicy po sortowaniu ");

for (i=0;i<8;i++)

    printf(" % d ",tab[i]);

printf("\nrozmiar to : %d",sizeofTab);

if(sizeofTab%2==0)

{

    sizeHalf = sizeofTab/2;

    med=(tab[sizeHalf]+tab[sizeHalf-1])/2;

```

```

    }

    if(sizeofTab%2==1)
    {
        sizeHalf= sizeofTab/2;
        med=tab[sizeHalf+1];
    }

    printf("\nMediana wynosi : %d",med);

    printf("\n");

    system("pause");
}

```

Widok z terminala :

```

tab[0]=9
tab[1]=5
tab[2]=-2
tab[3]=7
tab[4]=-4
tab[5]=15
tab[6]=-20
tab[7]=100

Stan tablicy po sortowaniu  -20  -4  -2   5   7   9   15  100
rozmiar to : 8
Mediana wynosi : 6
Press any key to continue . . .

```