Temat zajęć: **Tablice jednowymiarowe w języku C**

Autor: mgr inż. Sławomir Samolej

Zagadnienie 1. (Tablice liczbowe jednowymiarowe)

Tablica jest ciągiem identycznych elementów znajdujących się jeden za drugim w pamięci. Podstawowe informacje o tablicy to jej długość i typ danych jej elementów, np.:

```
int tab1[30]; // 30-elementowa tablica elementów typu int
char tekst[10]; // 10-elementowa tablica elementów typu char (tablica tekstowa)
```

Odwołać się do elementu tablicy można przez indeks do tego elementu np.:

```
int tab2[4]={2,5,6,7};
int a, b;
a=tab2[0]; // zmiennej a przypisz zawartość pierwszego elementu tablicy tab2
b=tab2[3]; // zmiennej b przypisz zawartość ostatniego elementu tablicy tab2
tab2[2]=45; // elementowi tablicy o indeksie 2 nadaj wartość 45
```

Uwaga: **Indeksowanie tablicy odbywa się zawsze od 0!** Ostatni element ma indeks równy (rozmiar tablicy) –1!

Zadania:

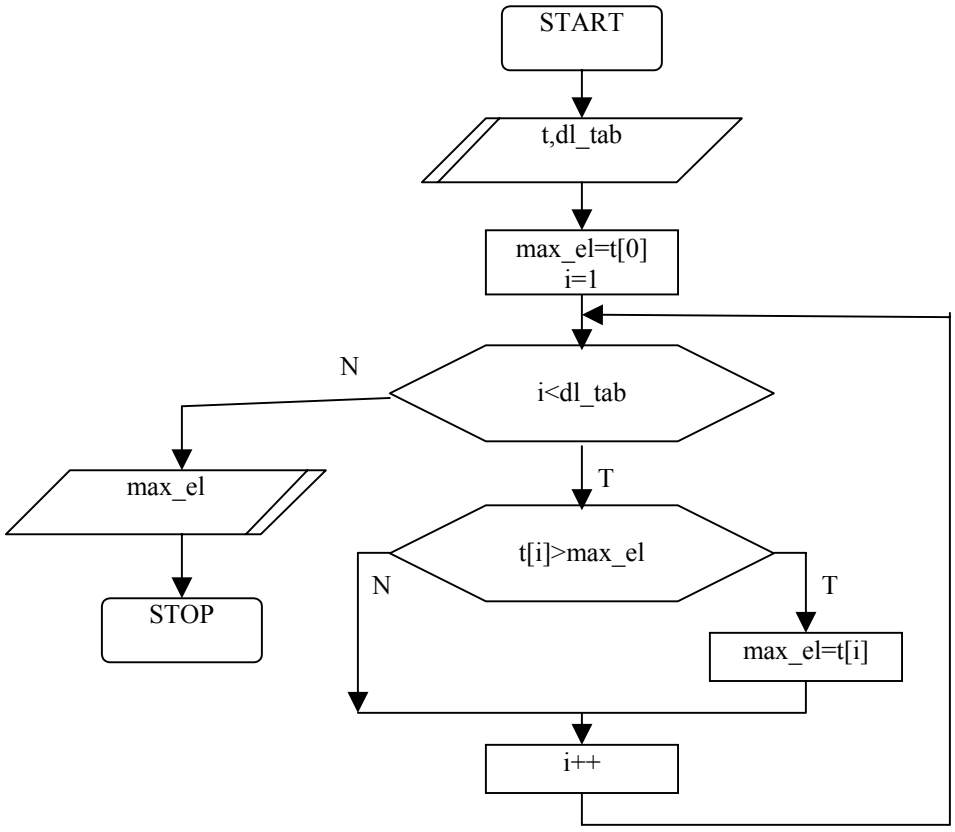
Zadanie 1.

Dany jest program:

```
#include <stdio.h>

#define DL_TAB 10  // definicja stałej DL_TAB

int tab[DL_TAB];      // globalna tablica typu int o dlugosci DL_TAB

int max(int t[],int dl_tab);

void main(void)
{
        int max_el,i;
        printf("Podaj 10 liczb stalopozycyjnych:\n");
        for(i=0;i<DL_TAB;i++)
        {
                printf("tab[%d]=",i);
                scanf("%d",&tab[i]);
        }
        printf("Zawartosc tablicy:\n");
```

```
        for(i=0;i<DL_TAB;i++)
                printf("tab[%d]=%d\n",i,tab[i]);
        max_el=max(tab,DL_TAB);
        printf("Najwiekszy element tablicy wynosi: %d\n",max_el);
}

int max(int t[],int dl_tab)
{
        int i, max_el=t[0];

        return max_el;
}
```

Uzupełnić funkcję „max” w taki sposób, aby przeszukała ona tablicę „t” o rozmiarze „dl_tab”, znalazła największy element tej tablicy, a następnie zwróciła go. Proponowany algorytm działania funkcji:



Zadanie 2.

Dany jest program:

```
#include <stdio.h>

#define DL_TAB 5    // definicja stałej DL_TAB

int tab[DL_TAB];      // globalna tablica typu int o dlugosci DL_TAB

float sr(int t[],int dl_tab);

void main(void)
{
        float srednia;
        int i;
        printf("Podaj 5 liczb stalopozycyjnych:\n");
        for(i=0;i<DL_TAB;i++)
        {
                printf("tab[%d]=",i);
                scanf("%d",&tab[i]);
        }
        printf("Zawartosc tablicy:\n");
        for(i=0;i<DL_TAB;i++)
                printf("tab[%d]=%d\n",i,tab[i]);
        srednia=sr(tab,DL_TAB);
        printf("Srednia z wszystkich elementow tablicy wynosi: %f\n",srednia);
}

float sr(int t[],int dl_tab)
{
        int i;
        float srednia;

        return srednia;
}
```

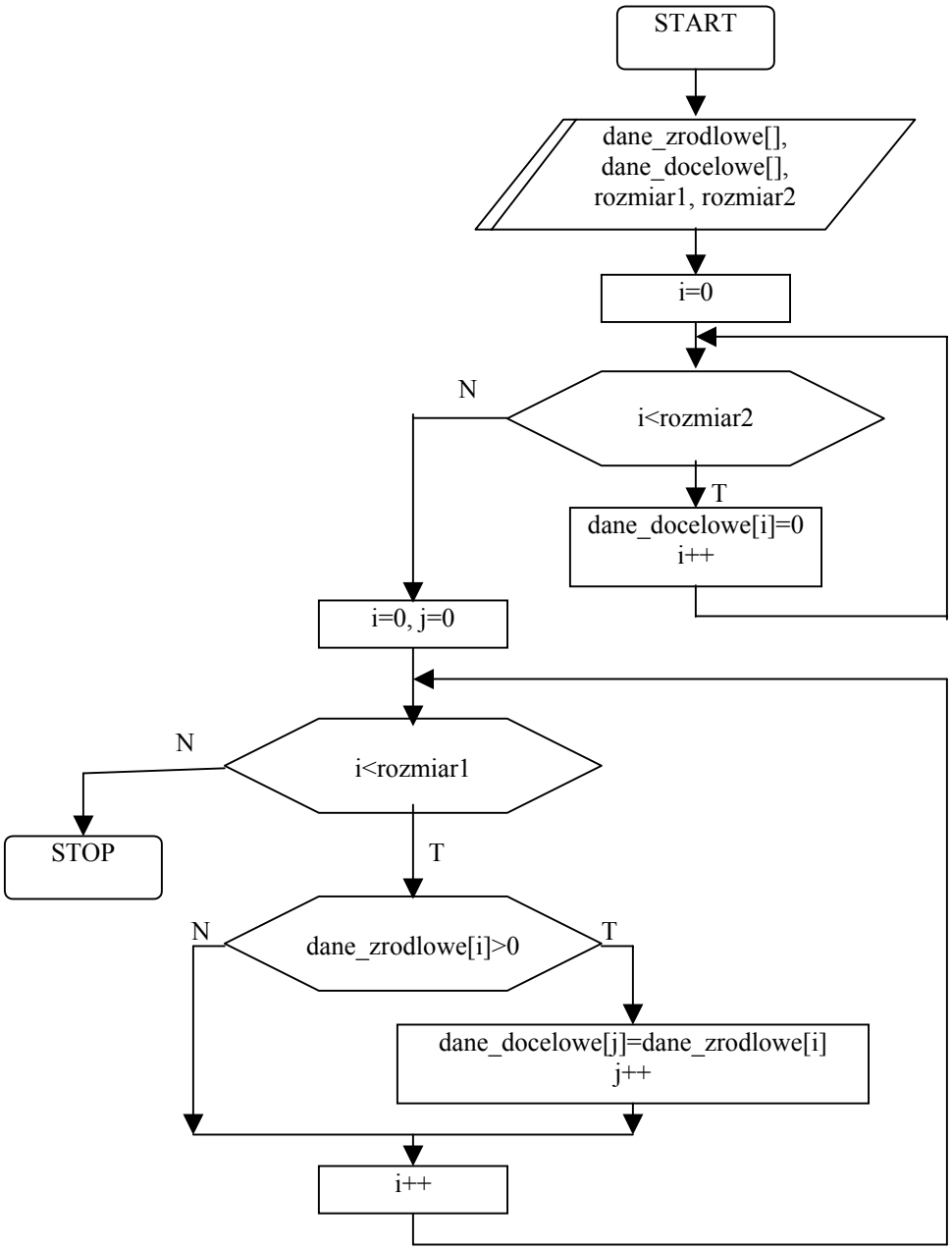
Uzupełnić funkcję „sr” w taki sposób, aby wyliczyła i zwróciła średnią arytmetyczną z wartości zawartych w tablicy „t” o długości „dl_tab”.

Zadanie 3.

Dany jest program:

```
#include <stdio.h>

int t1[6]={1,-3,4,-5,3,2};
int t2[10];

void kopiuj(    int dane_zrodlowe[],int rozmiar1,
                int dane_docelowe[],int rozmiar2
                );

void main(void)
{
        int i;
        printf("Dane zrodlowe (przed przetwarzaniem):\n");
        for(i=0;i<6;i++) printf("dane_zrodlowe[%d]=%d\n",i,t1[i]);
        printf("Dane docelowe (przed przetwarzaniem):\n");
        for(i=0;i<10;i++) printf("dane_docelowe[%d]=%d\n",i,t2[i]);
        kopiuj(t1,6,t2,10);
        printf("Dane zrodlowe (po przetwarzaniu):\n");
        for(i=0;i<6;i++) printf("dane_zrodlowe[%d]=%d\n",i,t1[i]);
        printf("Dane docelowe (po przetwarzaniu):\n");
        for(i=0;i<10;i++) printf("dane_docelowe[%d]=%d\n",i,t2[i]);
}

void kopiuj(    int dane_zrodlowe[],int rozmiar1,
                int dane_docelowe[],int rozmiar2)
{

}
```

Uzupełnić funkcję „kopiuj”. Zadaniem funkcji jest przeniesienie wszystkich dodatnich elementów z tablicy dane_zrodlowe o rozmiarze rozmiar1 na kolejne miejsca w tablicy dane_docelowe o rozmiarze rozmiar2. Ewentualne wolne miejsca w tablicy dane_docelowe mają zostać wypełnione 0. Przykład:
Przed przetworzeniem tablice mają postać:
int t1[6]={1,-3,4,-5,3,2};
int t2[10]={};
Po wykonaniu funkcji kopiuj tablice powinny mieć postać:
int t1[6]={1,-3,4,-5,3,2};
int t2[10]={1,4,3,2,0,0,0,0,0,0};
Poniżej podano algorytm rozwiązania zadania:


START
dane_zrodlowe[],
dane_docelowe[],
rozmiar1, rozmiar2
i=0
N
i<rozmiar2
T
dane_docelowe[i]=0
i++
i=0, j=0
N
i<rozmiar1
STOP
T
N
dane_zrodlowe[i]>0
T
dane_docelowe[j]=dane_zrodlowe[i]
j++
i++

Zagadnienie 2. (Tablice tekstowe)

Tablice o elementach typu char mogą przechowywać teksty. Tekst w języku C jest ciągiem znaków (typu char) zakończonych liczbą 0. Liczbę 0 w tablicach tekstowych często koduje się przy pomocy specjalnego znaku: '\0'. Jeśli chcemy zachować w tablicy o elementach typu char pewien tekst, możemy zainicjalizować tablicę w pamięci, a następnie przypisać jej stałą tekstową, np.:
char tekst1[100]="To jest tekst";

W pamięci w poszczególnych elementach tablicy zapisane zostaną kolejne znaki tekstu, a na koniec '\0':
tekst1[0]=='T', tekst1[1]=='o', tekst1[2]==' ',..., tekst1[12]='t', tekst1[13]=='\0',

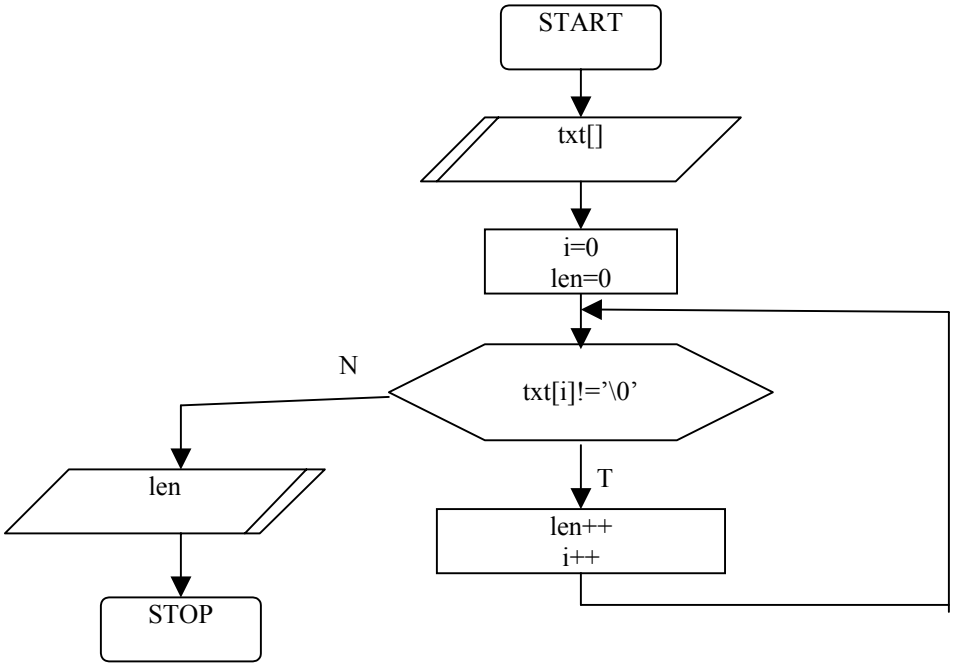
Przy inicjalizacji tablicy należy zwrócić uwagę na rozmiar tablicy. Rozmiar powinien być wystarczający do przechowania tekstu. W przykładowej tablicy tekst1 o rozmiarze 100 elementów, zapamiętany został tekst o długości 13 znaków (w długości tekstu nie uwzględnia się znaku '\0' na końcu każdego tekstu). Podczas przetwarzania tekstów z reguły nie uwzględnia się faktycznej długości tablicy, a raczej długość tekstu w niej zawartego.

Zadanie 4.

Dany jest program:

```
#include <stdio.h>

char tekst[100];

int str_len(char txt[]);

void main(void)
{
	int len;
	printf("Podaj dowolny tekst (do 80 znakow):\n");
	gets(tekst);                                    // funkcja pobierajaca jedna linie tekstu
	len=str_len(tekst);
	printf("Tekst: \"%s\" \nma dlugosc: %d\n",tekst,len);
}

int str_len(char txt[])
{
}
```

Uzupełnić funkcję str_len w taki sposób, aby obliczała i zwracała długość tekstu zawartego w tablicy tekstowej przekazanej przez parametr „txt". Poniżej podano propozycję algorytmu:



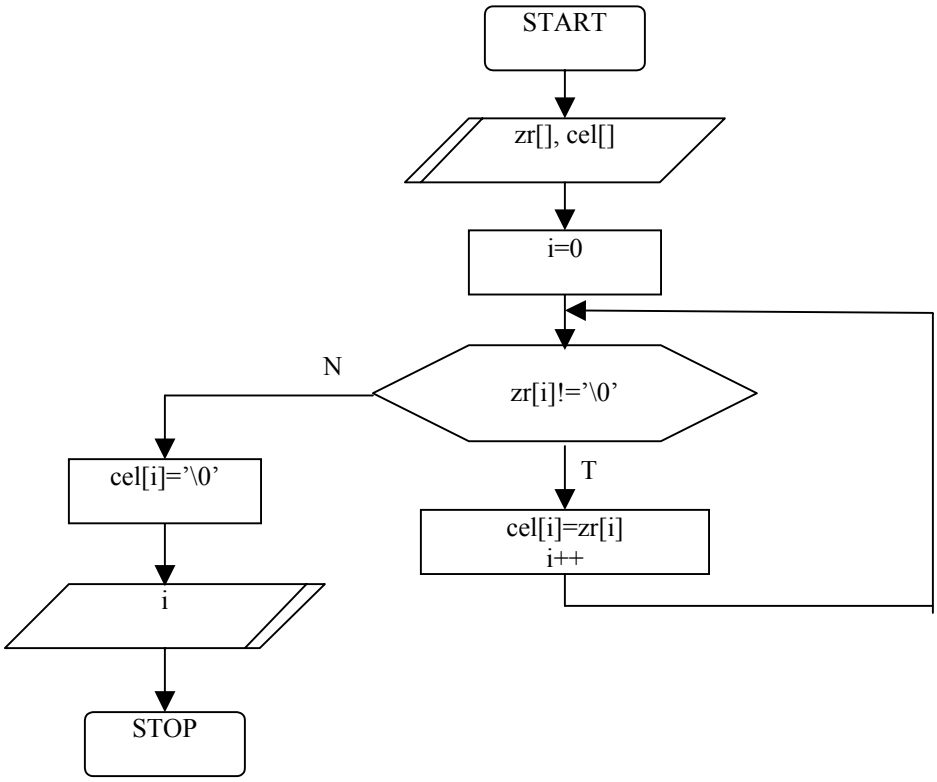
Zadanie 5.

Dany jest program:

```
#include <stdio.h>

char tekst1[100]="To jest tekst do kopiowania";
char tekst2[100];

int str_cpy(char zr[], char cel[]);

void main(void)
{
	int len;
	printf("Zawartosc tablicy zrodlowej (przed przetwarzaniem):\n");
	printf("%s\n",tekst1);
	printf("Zawartosc tablicy docelowej (przed przetwarzaniem):\n");
	printf("%s\n",tekst2);
	// Przetwarzanie:
	len=str_cpy(tekst1,tekst2);
	printf("Zawartosc tablicy zrodlowej (po przetwarzaniu):\n");
	printf("%s\n",tekst1);
	printf("Zawartosc tablicy docelowej (po przetwarzaniu):\n");
	printf("%s\n",tekst2);
}
int str_cpy(char zr[], char cel[])
```

```
{
}
```

Uzupełnić funkcję str_copy w taki sposób, aby dokonywała kopiowania tekstu z tablicy źródłowej (zr) do tablicy docelowej (cel). Dodatkowo funkcja powinna zwracać ilość przekopiowanych znaków. Poniżej podano propozycję algorytmu funkcji:



Zadanie 6.

Dany jest program:

```
#include <stdio.h>

char tab[60];

int atoi(char s[]);

void main(void)
{
        int liczba;
        printf("Podaj liczbe calkowitą nieujemna:\n");
        gets(tab);
```

```
        printf("Wczytana liczba w postaci tablicy znakow: %s\n",tab);
        liczba=atoi(tab);
        printf("Wczytana liczba w postaci liczby calkowitej: %d\n",liczba);
}

int atoi(char s[])
{
}
```

Napisać funkcję “atoi”, która przekształca liczbę zapisaną w postaci tablicy tekstowej na liczbę zapisaną jako dana typu całkowitego. Poniżej zaproponowano algorytm funkcji: