Laboratorium 3.

Zad. 1.

Mając deklarację tablicy:

```
double x[5];
```

Powiedz, jaka jest:

a) nazwa tablicy
b) typ tablicy
c) deklarowany rozmiar tablicy
d) dopuszczalny zakres indeksów przyjmowanych przez indeks tablicy
e) nazwa elementu tablicy (jednej zmiennej indeksowanej)

Zad. 2.

Prawda czy fałsz?

Do elementów tablicy char `a[5]` odwołujemy się następująco:

`a[0], a[1], a[2], a[3], a[4] i a[5].`

Zad. 3.

Wskaż błędy w poniższych instrukcjach. Jaka będzie reakcja kompilatora?

a)
```
int a[4] = {1,2,3,4,5};
```

b)
```
int a[] = {1,2,3,4};
```

c)
```
const int ROZMIAR = 4;
int a[ROZMIAR];
```

d)
```
cout << "Ile elementow?" << endl;
cin >> ile;
int t[ile];
```

e)
```
int t[10];
for (int i=1; i<=10; i++) t[i]=3*i;
```

Zad. 4.

Zadeklaruj tablicę o pięciu elementach i zainicjuj ją wartościami:

`.1, .2, .3, .4 i .5.`

Zad. 5.

Poniższy fragment programu zawiera instrukcje, które podwajają wartości w tablicy `punkty`. Znajdź błąd:

```
int i, punkty[16];
i=0;
while (i <= 16) {
      wyniki[i] *= 2;
      i++;
}
```

Zad. 6.

Załóżmy, że elementy tablicy są posortowane rosnąco:

`a[0]≤ a[1] ≤  a[2] ≤ ..`

Program ma kontrolować czy tablica jest posortowana. Czy poniższy fragment programu jest poprawny?

```
const int MAXEL = 10;
double a[MAXEL];
...
for (int i=0; i < MAXEL-1; i++)
  if (a[i] > a[i+1])
  cout << "elementy o indeksach " << i << " oraz " << i+1
    << "sa w niewlasciwej kolejnosci" << endl;
```

Zad. 7.

Napisz funkcję, która przypisuje pierwszym `n` elementom tablicy wartości od 1 do `n`.

Zad. 8.

Napisz funkcję, która:

- wczytuje n liczb do tablicy (n jest podawane przez użytkownika i zwracane jako wartość funkcji);
- drukuje je w odwrotnej kolejności (parametry tablica, n).

Zad. 9

Co zlicza funkcja:

```
int fun(int a[], int n) {
      int i, licznik=0;
      for (i=0; i<n; i++) {
```

```
        if (a[i] != 0) break;
        licznik++;
        }
    return licznik;
    }
```

Praca domowa

Zad. 10.

Napisz funkcję, która wyznacza średnią wartość elementów tablicy, parametr `n` określa liczbę elementów w tablicy.

```
double Srednia(int a[], int n);
```

Zad. 11.

Napisz funkcję, która oblicza następującą sumę elementów tablicy:

a[0] – a[1] + a[2] – a[3] + … – a[n-1].

Zad. 12.

Napisz funkcję, która przypisuje elementom tablicy następujące wartości:

`1,2,..., n-1, n, n-1, ..., 2, 1.`

Schemat funkcji:

```
void T (int a[], int n)
{
int i=1;
while (i<=n) {
    a[....] = i;
    a[....] = i;
    i++;
    }
}
```

Zad 13.

Uzupełnij funkcję przedstawioną poniżej. Funkcja ta porządkuje elementy w tablicy tak, aby wszystkie wartości ujemne znajdowały się z lewej strony wartości dodatnich.

```
void Przestaw(double a[], int rozmiar) {
int i=0, j=rozmiar-1;
double robocza;

    while (i<j) {
      if (a[i]<0) // Pomiń
       ......................
      else if ( ........... ) // Pomiń
       ......................
      else { // Przestaw)
```

```
        ......................
        ......................
        ......................
        }
                }
}
```

Zad. 14.

Napisz funkcję, która sprawdza, czy w tablicy występuje przynajmniej jedna para liczb jednakowych.

Zad. 15.

Napisz funkcję, która przesuwa cyklicznie elementy w tablicy o rozmiarze `n` o `d` pozycji:

```
void Przesun(int v[], int n, int d);
```

Dodatnia wartość `d` oznacza przesunięcie do przodu, ujemna – do tyłu.

Przykład:

```
int v[5]={1,4,9,16,25};
```

Po wywołaniu funkcji `Przesun(v,5,-2)` zawartość tablicy `v` to: `9,16,25,1,4.`

Zad. 16.

Napisz funkcję, która wyznacza następujące sumy:

```
s[0]=a[0]
s[1]=a[0] + a[1]
...
s[rozmiar-1] = a[0] + a[1] + ... + a[rozmiar-1]
```

Prototyp funkcji:

```
void Suma(double a[], double s[], int n);
```

Zad. 17.

Napisz funkcję, która włącza do tablicy zawierającej uporządkowane liczby nowa liczbę, bez naruszania istniejącego porządku. Jeśli liczba jest większa od największej liczby w tablicy, pomiń ją.

a) Załóż, że w tablicy liczby nie powtarzają się.

b) Załóż, że w tablicy liczby mogą się powtarzać.

c) Załóż, że tablica jest cała wypełniona, wprowadzanie nowej liczby spowoduje usunięcie ostatniej liczby.

d) Załóż, że tablica nie jest do końca wypełniona. Dopóki jest miejsce, przesuwaj liczby. Gdy miejsca zabraknie, postępuj tak jak w punkcie c.