lekcja 6
Tablice

Deklaracja tablicy – Inicjowanie tablicy – zapis i odczyt komórek – tablica w parametrze funkcji

Deklaracja tablicy

Po co komu tablice? Jeśli w programie używamy 2 zmiennych, to nie ma problemu. Ale jeśli musimy posłużyć się 1000 zmiennych i na ich podstawie coś tam obliczyć, to pojawiają się problemy. Jak je zadeklarować i jak potem nie pogubić się w gąszczu tylu nazw! Tablice umożliwiają tworzenie struktur podobnych do pokratkowanej kartki – do każdej kratki można wpisać jakieś dane. Każda z kratek opisana jest tzw. indeksem, czyli numerem komórki. Komórki tablic w C++ numerowane są od zera!

typ nazwa [liczba elementów];

typ nazwa [liczba elementów] [liczba elementów] …;

dla tablic wielowymiarowych

Przykłady prostych tablic

int T[10];	tablica T o 10 elementach typu int numerowanych 0..9
char ZNAKI[20];	tablica ZNAKI o 20 elementach typu char numerowanych 0..19
int TAB[10][10];	tablica TAB o 100 elementach typu int dwuwymiarowa – 10 wierszy i 10 kolumn

Zapis do tablicy

T[0] = 1;	*T[9]=10;*
ZNAKI[1] = 'A';	pojedyncze znaki w apostrofach, teksty w cudzysłowie
TAB[0][0] = 1;	*TAB[9][7]=63;*

Deklaracja i inicjowanie tablicy

Każda komórka tablicy po zadeklarowaniu zawiera tzw. dane nieokreślone i dlatego (podobnie jak to czyniliśmy ze zmiennymi) powinny być zainicjowane - zerujemy je lub wstawiamy konkretne wartości.

Inicjowanie podczas deklarowania

int T[10] = {0,1,2,3,4,5,6,7,8,9};

char ZNAKI[20]={'a','l','a',' ','m','a',' ','k','o','t','a'}

Zerowanie za pomocą pętli (numerujemy od 0 do 9 – nie ma komórki o numerze 10)

for (int i=0;i<10;i++) T[i]=0;

Zadanie – Losowanie liczb

Wypełnij tablicę liczbami losowymi.

Losowane są liczby z przedziału 0..32768. Zakres losowanych liczb zależy to od rodzaju użytego kompilatora.

```
LOS[0]=41
LOS[1]=18467
LOS[2]=6334
LOS[3]=26500
LOS[4]=19169
LOS[5]=15724
LOS[6]=11478
LOS[7]=29358
LOS[8]=26962
LOS[9]=24464
```

losowe z przedziału 0..9	`LOS[i]=rand() % 10;`
losowe oceny 1..6	`LOS[i]=rand() % 6 + 1`

```
const int N=10;
int LOS[N];
for (int i=0;i<N;i++)
  LOS[i]=rand();
for (int i=0;i<N;i++)
  cout<<"LOS["<<i<<"]="<<LOS[i]<<endl;
```

Zwróć uwagę, że za każdym razem losowane są **te same liczby!** Rozwiązaniem jest zainicjowanie tzw. „maszyny losującej" – powiązanie początku losowania z czasem systemowym komputera –biblioteka **time** i funkcje **srand** i **time**.

```
#include <ctime>
...
srand(time(NULL));
```

Zadanie - Funkcja losująca

Napisz funkcję, która wylosuje liczbę całkowitą z przedziału POCZĄTEK..KONIEC

```
int LOSUJ(int poc, int kon){
  return rand()%(kon-poc+1)+poc;
}
```

Zadanie – Liczby parzyste

Zadeklaruj tablicę PARZ[100] o elementach typu int i wpisz do niej 100 pierwszych liczb parzystych. Następnie wypisz elementy tej tablicy, od tyłu, po 10 w każdym wierszu.

```
198 196 194 192 190 188 186 184 182 180
178 176 174 172 170 168 166 164 162 160
158 156 154 152 150 148 146 144 142 140
138 136 134 132 130 128 126 124 122 120
118 116 114 112 110 108 106 104 102 100
 98  96  94  92  90  88  86  84  82  80
 78  76  74  72  70  68  66  64  62  60
 58  56  54  52  50  48  46  44  42  40
 38  36  34  32  30  28  26  24  22  20
 18  16  14  12  10   8   6   4   2   0
```

- Inicjujemy maszynę losującą (generator liczb pseudolosowych)
- Tablica PARZ, stuelementowa, komórki od 0 do 99
- Wypełniamy tablicę od zera, mnożąc kolejną naturalną *2
- Wypisując na ekranie numerujemy w pętli wstecz od 99
- Jeżeli licznik pętli jest podzielny przez 10, to nowy wiersz

```
#include <iomanip>
#include <ctime>
...
srand(time(NULL));
int PARZ[100];
for (int i=0;i<100;i++)
  PARZ[i]=i*2;
for (int i=99;i>=0;i--){
  cout.width(4);
  cout << PARZ[i];
  if (i%10==0) cout<<endl;
}
```

Tablica jako parametr funkcji

Tablica przekazywana za pomocą parametru **nie musi mieć podanej konkretnej wielkości** – wystarczą same nawiasy kwadratowe. Wyjątkiem są tablice wielowymiarowe, z przekazywaniem których, zwłaszcza tych dużych, mogą pojawić się problemy.

W C++ przyjęto domyślnie, że **tablice ZAWSZE będą przekazywane przez referencję** – zawsze działamy na rzeczywistej tablicy (wyjątkiem są duże tablice).

Wszystkie działania na tablicach (zwłaszcza pętle) muszą „wiedzieć", kiedy zakończyć działanie. Niestety w C++ nie ma gotowej funkcji, która podawałaby liczbę komórek tablicy i dlatego tworząc wszelkiego rodzaju funkcje, najlepiej prócz samej tablicy **należy podawać zakres komórek do przeszukania**!

```
//MINIMUM
int Tmin(int poc, int kon, int t[]){
  int m=t[poc];
  for (int i=poc+1;i<=kon;i++)
    if (t[i]<m) m=t[i];
  return m;
}
//SUMA
int Tsuma(int poc, int kon, int t[]){
  int s=0;
  for (int i=poc;i<=kon;i++)
    s=s+t[i];
  return s;
}
...
//LOSOWANIE
srand(time(NULL));
const int N=10;
int LOS[N];
for (int i=0;i<N;i++) LOS[i]=rand();
cout << Tmin(0,9,LOS) << endl;
cout << Tsuma(0,9,LOS) << endl;
```

Zadanie – Minimum-Suma

Wypełnij tablicę liczbami losowymi i wyszukaj najmniejszą (największą) oraz wylicz sumę wylosowanych liczb. Napisz trzy funkcje Tmin (Tmax) i Tsuma, których parametrem jest tablica z liczbami.

Podczas wyszukiwania minimalnej i maksymalnej liczby, porównujemy w pętli komórki z tablicy z tymczasową zmienną: MIN lub MAX, która powinna być zainicjowana – najlepiej pierwszym elementem z tablicy.

Zadanie - Macierz

Macierze wykorzystywane są w matematyce, m.in. do rozwiązywania równań liniowych. Macierz jest tablicą dwuwymiarową. Jedną z ważniejszych operacji na macierzach jest transpozycja, czyli zamiana wierszy z kolumnami, np. tablica 4x5 zamienia się w tablicę 5x4.

1) Zadeklaruj dwie tablice A 4x5 i B 5x4 typu int.
2) Wypełnij tablicę A liczbami losowymi z przedziału -9..9.
3) Wyświetl na ekranie tablicę A.
4) Dokonaj transpozycji macierzy A do B.
5) Wyświetl tablicę B na ekranie.

```
 9  -2  -5   0
 2  -5  -4  -1
-6   7  -7   7
 0  -8   8  -8
 0  -1  -4   5

 9   2  -6   0   0
-2  -5   7  -8  -1
-5  -4  -7   8  -4
 0  -1   7  -8   5
```

```
int A[5][4];
int B[4][5];
//losowanie
for (int w=0;w<5;w++){
for (int k=0;k<4;k++){
  A[w][k]=LOSUJ(-9,9);
}}
//wysietlanie
for (int w=0;w<5;w++){
for (int k=0;k<4;k++){
  cout.width(4);
  cout<<A[w][k];
  } cout<<endl;
} cout<<endl;
//transpozycja
for (int w=0;w<5;w++){
for (int k=0;k<4;k++){
  B[k][w]=A[w][k];
}}
//wyswietlanie
for (int w=0;w<4;w++){
for (int k=0;k<5;k++){
  cout.width(4);
  cout<<B[w][k];
  } cout<<endl;
}
```

Zadanie - Sortowanie naiwne

W tablicy znajdują się losowe liczby. Należy je uporządkować wykorzystując tzw. sortowanie naiwne według poniższego algorytmu:

1. ***szukamy** minimalnej liczby w zakresie pierwsza .. ostatnia komórka tablicy*
2. ***zamieniamy** liczby w komórkach: znaleziona liczba minimalna – pierwsza*
3. *zwiększamy numer pierwszej komórki o 1 i powtarzamy proces o punktu 1*

Potrzebne będą dwie funkcje pomocnicze: SZUKAJ i ZAMIANA.

Funkcja pomocnicza ZAMIANA zamienia liczby w dwóch zmiennych „a" i „b" – konieczna jest jeszcze jedna zmienna – tzw. zmienna pomocnicza. Parametry funkcji przekazywane są przez tzw. **referencję** – po wywołaniu funkcji w parametrach będą zamienione liczby.

```
void ZAMIANA (int &a, int &b) {
  int p=a;  a=b; b=p;
}
int SZUKAJ(int poc, int kon, int t[]){
  int m=t[poc];
  int min=poc;
  for (int i=poc+1;i<=kon;i++)
      if (t[i]<m) {
         m=t[i];
         min=i;
       }
  return min;
}
```

Funkcja pomocnicza SZUKAJ wyszukuje element minimalny, ale zwraca numer komórki, w której znajduje się minimalna wartość tablicy.

Funkcja działa według algorytmu:

1. przeglądamy tablicę *t[]* w pętli od *poc* do *kon*
2. wartość minimalną zapisujemy w zmiennej *m*, pozycję wartości minimalnej w *min*
3. funkcja zwraca numer minimalnego elementu w tablicy

Funkcja główna SORTUJ wykorzystuje dwie funkcje pomocnicze. Sortujemy całą tablicę, dlatego jako parametrem jest tablica i ilość elementów tablicy. Funkcja zwraca posortowaną tablicę, ponieważ tablice przekazywane są jako referencje. Algorytm funkcji:

1. zmienna *ile* zawiera liczbę elementów tablicy, dlatego pętla kręci się w zakresie *0..ile-1*
2. funkcja ZAMIANA zamienia miejscami dwie komórki tablicy (jeśli jest to konieczne) komórkę z liczbą minimalną (funkcja Tmin) i pierwszą badaną komórkę
3. funkcja zwraca posortowaną tablicę

```
void SORTUJ(int ile, int t[]){
  for (int p=0;p<ile-1;p++)
    ZAMIANA(t[SZUKAJ(p,ile-1,t)], t[p]);
}
...
const int N=10;
int LOS[N];
//losowanie
for (int i=0;i<N;i++) LOS[i]=rand();
for (int i=0;i<N;i++)
  cout<<"LOS["<<i<<"]="<<LOS[i]<<endl;
//sortowanie
SORTUJ(10,LOS);
for (int i=0;i<N;i++)
  cout<<"LOS["<<i<<"]="<<LOS[i]<<endl;
```

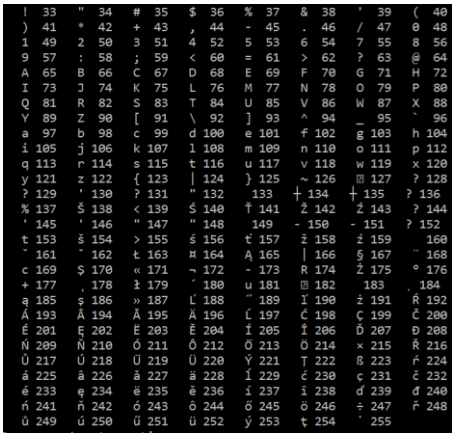
Bardziej „zrozumiała" wersja sortowania –wyszukiwanie minimum i zamiana liczb w osobnych wierszach.

Uwaga. Jeśli przekazywanie tablicy jako parametr funkcji sprawia problemy (zbyt duża, albo wielowymiarowa), to może (co nie jest rozwiązaniem „edukacyjnym") w funkcji odwołuj się bezpośrednio do niej!

```
void SORTUJ(int ile, int t[]){
  int min;
  for (int p=0;p<ile-1;p++){
    min=SZUKAJ(p,ile-1,t);
    ZAMIANA(t[min], t[p]);
  }}
```

Zadania

1) *Do tablicy znakowej wpisz kolejne litery alfabetu ASCII od numeru 33 do 255. Wypisz je na ekranie.*



2) *Zmodyfikuj funkcję SORTUJ tak, aby sortowała w kolejności malejącej. Wylosuj liczby do tablicy, wypisz nieposortowane, posortuj i wypisz posortowane.*

```
0 1 0 5 6 7 2 4 0 6 7 6 4 5 0 0 8 5 4 7
8 7 7 7 6 6 6 5 5 5 4 4 4 2 1 0 0 0 0 0
```

3) *Do tablicy dwuwymiarowej MN[10][10] wpisz tabliczkę mnożenia w zakresie 1..10. Wypisz zawartość tablicy na ekranie.*

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

4) *Tablica **trój**wymiarowa OCENY zawiera losowe oceny semestralne dla 5 klas, 7 przedmiotów i 10 uczniów OCENY[5][7][10].*
   a) wyzeruj tablicę OCENY
   b) rozlosuj oceny (1..6) do tablicy
   c) wypisz wszystkie oceny klasami
   d) wylicz średnią z ocen dla wybranej klasy, ucznia, przedmiotu
   e) wypisz na ekranie oceny wybranej klasy, ucznia, przedmiotu – wybierz z klawiatury

5) *Wylosuj liczby do tablicy, a następnie wypisz osobno parzyste i nieparzyste*

6) *Wylosuj liczby do tablicy. Wylicz różnicę między MAX i MIN*

7) *Wylosuj liczby do tablicy. Wylicz średnią i element najbliższy średniej*