

Sprawozdanie, tablice, pętle

Tablice – służą do gromadzenia danych jednego typu jedna po drugiej w pamięci. Deklaracja: typ nazwa[liczba_komórek]. Istnieje możliwość stworzenia tablicy dwuwymiarowej ([][]), która jest odpowiednikiem macierzy. Istnieją 3 rodzaje pętli, while – wykonuje się do momentu niespełnienia warunku. Do ... while, wykonuje się **przynajmniej** raz, dalej jak while. For – konstrukcja – (inicjalizacja (najczęściej) iteratorów ; warunek, jeżeli niespełniony to przerwij pętlę; instrukcje wykonywane po każdej iteracji, najczęściej in/dekrementacja iteratorów.

Simple_loop:

wyszukiwanie max elementu w tablicy:

```
int max = tablica_int[0];
for (int i = 1; i < ROZMIAR_TABLICY; i++){
    if (max < tablica_int[i])
        max = tablica_int[i];}
```

Zapis elementów losowych:

```
srand(time(NULL));
for (int i = 0; i < ROZMIAR_TABLICY2; i++)
    tablicaczarow[i] = (rand() % 93) + 33;
```

Szukanie zadanej wartości:

```
for (int i = 0; i < ROZMIAR_TABLICY2; i++){
    if (look_for==(int)tablicaczarow[i]){
        printf("wartość jest w %d el tab", i + 1);
        break;}}
```

Wszystkie wartości – bez break;

Obliczanie sumy:

```
int suma = 0;
for (int i = 0; i < ROZMIAR_TABLICY; i++)
    suma += tablica_int[i];
```

```
element maksymalny: 10
suma elementow tablicy: 55
```

Znalezienie średniej wartości:

```
int suma1 = 0;
for (int i = 0; i < ROZMIAR_TABLICY2; i++)
    suma1 += (int) tablicaczarow[i];
double sr = suma1/ROZMIAR_TABLICY2
```

```
srednia elementow tablicy: 78
podaj wartosc szukanego elementu (od 33 do 126):
67
Pierwszy znaleziony element:
Znaleziono wartosc 67 w 100 elemencie tablicy
```

Oblicz_PI

Zatrzymanie pętli w przypadkach krytycznych:

elementy szeregu są mniejsze od minimalnej wartości

```
if(suma_plus < SMALL_NUMBER || suma_minus <
SMALL_NUMBER)
```

```
break;
```

Użycie różnych pętli:

while – przy podaniu adekwatnej dokładności aby warunek się nie spełnił pętla nie zostanie wykonana. for - możliwy jest brak wejścia w pętlę, poza tym różnic się będzie zapis, poszczególne operacje można umieścić w warunkach

for - możliwe jest brak wejścia w pętlę, poza tym różnic się będzie zapis, poszczególne operacje można będzie umieścić argumentach funkcji

Różnica kolejnych wartości PI jest mniejsza od min. wart:

```
if(aktualnePI - poprzedniePI < SMALL_NUMBER )
break;
```

```
przerwanie petli w 843
osiagnieto maksymalna dokladnosc
PI obliczone: 3.140997886657715, PI z biblioteki matematycznej: 3.141592653589793
Zalozona dokladnosc: 0.000000001000000, rzeczywisty blad: 0.000594766932078
```

Oblicz_e

Sam program służący do obliczania liczby e ma konstrukcję podobną do liczby PI, jeżeli brać pod uwagę konstrukcje

sterujące oraz zabezpieczenia. Matematycznie korzystamy z następującej sumy: $e = \sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{\infty!}$

e dla 10 elementów sumy:

```
e obliczone: 2.718281525573192, e z biblioteki matematycznej: 2.718281828459045
```

