**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;

**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 pętle, poza tym różnic się będzie zapis, poszczególne operacje można umieścić w warunkach

**for** - możliwe jest brak wejścia w pętle, poza tym różnić się

będzie zapis, poszczególne operacje można będzie umieścić argumentach funkcji

Obliczanie sumy:

int suma = 0;

for (int i = 0; i < ROZMIAR\_TABLICY; i++)

suma += tablica\_int[i];



Znalezienie średniej wartości:

int suma1 = 0;

for (int i = 0; i < ROZMIAR\_TABLICY2; i++)

suma1 += (int) tablicaczarow[i];

double sr = suma1/ROZMIAR\_TABLICY2

Obraz zawierający tekst

Opis wygenerowany automatycznie

Różnica kolejnych wartości PI jest mniejsza od min. wart:

if(aktualnePI - poprzedniePI < SMALL\_NUMBER ) break;

