

## WSKAŹNIKI

Cztery możliwości jakie dają nam wskaźniki:

1. Dynamiczne rezerwowanie i zwalnianie obszarów Pamięci.
2. Zwiększenie szybkość Zapisu i/ odczytu komórek w tablicy.
3. Dawanie funkcjom do pracy oryginałów zmiennych z programu wywołującego.
4. Możliwość współpracy z urządzeniem zewnętrznym, ( np. Miernik temperatury )

W języku C/C++ możemy używać zmiennych wskaźnikowych (lub wskazujących, ang. *pointer, pointing variables*), których wartościami są wskaźniki zawierające adres pewnego obiektu mogącego być dowolną zmienną, strukturą lub obiektem pewnej klasy. Ogólna postać zmiennej wskaźnikowej jest następująca:

```
typ_danych *nazwa_wskaźnika;
```

gdzie \* jest operatorem pobierania spod adresu. Przykład:

```
int *pa;
```

informuje kompilator, że tak zadeklarowana zmienna wskaźnikowa (wskaźnik do zmiennej) pa wskazuje zmienną typu int.

Aby zmienna wskaźnikowa wskazywała adres zmiennej, należy użyć operatora adresu oznaczonego symbolem &. Przykład wykorzystania operatora adresu ilustruje następująca instrukcja:

```
pa = &a;
```

gdzie zmienna pa podaje teraz adres zmiennej a. Jeśli chcemy uzyskać dostęp do wartości zmiennej a za pomocą wskaźnika pa, to musimy posłużyć się operatorem pobrania:

```
b = *pa .
```

Operator pobrania podaje zmienną wskazywaną przez dany wskaźnik. Wykonanie podanej instrukcji przypisania spowoduje, że zmiennej b zostanie przypisana wartość a. Zatem dwie instrukcje:

```
pa = &a;
```

```
b = *pa;
```

są równoważne instrukcji:

```
b = a;
```

**Zadanie 1)** Napisz program, który sprawdza, czy instrukcje `pa = &a` i `b = *pa` są równoważne instrukcji `b = a`.

Język C/C++ oferuje nam możliwość wykonywania operacji na tablicach za pomocą wskaźników. Wynika to stąd, że sama nazwa tablicy podaje adres pierwszego elementu tablicy. Dla tablicy:

```
int tab[5];
```

dostęp do pierwszego elementu tablicy (o indeksie 0) jest możliwy zarówno przez `tab[0]`, jak i przez `*tab`. Wskaźnikowi `ptab`, który został zadeklarowany w postaci:

```
int *tab;
```

można przypisać adres pierwszego elementu tablicy na dwa sposoby:

```
ptab = &tab[0]; // przypisanie adresu pierwszego elementu tablicy
```

```
ptab = tab; // przypisanie adresu pierwszego elementu tablicy
```

**Zadanie 2)** Napisz program, który ilustruje możliwość wykonywania operacji na tablicach za pomocą wskaźników.

**Zadanie 3)** Napisz program, który nadaje elementom zadeklarowanej wcześniej tablicy `int tab[5]` wartość 5. Wykonaj w programie te same czynności dla wyrażenia `tab + i`, gdzie jest ono adresem *i*-tego elementu tablicy.

Podpowiedź: wiemy, że aby nadać wszystkim elementom zadeklarowanej wcześniej tablicy na przykład wartość 5, musimy zastosować pętlę `for`: `for (i = 0; i < 5; i++) { tab[i] = 5; }` Ponieważ wyrażenie `tab + i` podaje adres *i*-tego elementu tablicy, możemy przypisać również wszystkim elementom wartość równą 5: `for (i = 0; i < 5; i++) { *(tab + i) = 5; }`

**Zadanie 4)** Napisz program, który oblicza sumę dwóch liczb całkowitych. Powinien on zawierać jeden podprogram: funkcję `dodaj_liczbe()`, do której parametry przekazywane są przez wskaźnik.

**Zadanie 5)** Napisz program, który oblicza sumę dwóch liczb całkowitych. Powinien on zawierać jeden podprogram: funkcję `dodaj_liczbe()`, do której parametry przekazywane są przez referencję.

Znacznie prostsze od stosowania wskaźników jest zadeklarowanie, że parametry formalne są typu referencyjnego. O parametrze formalnym powiemy, że jest typu referencyjnego, jeśli poprzedza go operator `&`. Tak zadeklarowane parametry nie zmieniają treści samej funkcji ani sposobu jej wywołania. Funkcja otrzymuje do dyspozycji zmienną będącą parametrem aktualnym i może zmieniać jej wartość. Takie przekazywanie parametrów nazywamy przekazywaniem przez adres. Treść samej funkcji nie ulega zmianie. Wywołanie funkcji również pozostaje bez zmian.

**Zadanie 6)** Napisz program, który dynamicznie przydziela pamięć do policzenia sumy liczb od 1 do 100, a następnie po zakończeniu obliczeń zwalnia przydzieloną pamięć.

### **Zadanie A)**

Stwórz zmienną `LICZBA`, zawierającą liczbę całkowitą (naturalną) 1234, następnie stwórz wskaźnik o nazwie `W`, i przypisz do niego adres alokacji zmiennej `LICZBA`, Wypisz ją na ekranie konsoli w zapisie dziesiętnym.

**A1)** Zmodyfikuj powyższy kod tak żeby wskaźnik `W` wypisał liczbę którą przechowuje.

### **Zadanie B)**

**B1)** Napisz funkcję liczącą ciąg fibonacciego.

**B2)** Skopiuj funkcję liczącą ciąg fibonacciego do osobnej funkcji i zmodyfikuj ją, tak aby wpisywała każdy element do tablicy. Obie funkcje umieść w funkcji `main()`.

**B3)** Zmodyfikuj funkcję z punktu B2), tak żeby rezerwowała przestrzeń tablicy na tyle elementów ile ma mieć ciąg.

### **Zadanie C)**

- Stwórz funkcję liczącą średnią arytmetyczną w tablicy.

- Zmodyfikuj tą funkcję tak żeby stosowała wskaźniki.
- Porównaj czas obliczenia obu funkcji.