

## Dynamiczny przydział pamięci C++

- Do przydzielania nowego bloku pamięci służy operator **new** (C++), np.:

```
int *wsk;
```

```
wsk = new int; // powoduje utworzenie nowego obiektu typu int
```

```
    // nie ma on nazwy, ale jego adres jest
```

```
    // przekazywany wskaźnikowi wsk
```

- Natomiast do zwolnienia dynamicznie przydzielonej pamięci służy operator **delete** (C++).

```
delete wsk; // powoduje likwidację powyższego obiektu (o ile
```

```
    // wskaźnik wsk nadal na niego pokazuje)
```

- Utworzenie tablicy:

```
float *tab;
```

```
tab = new float[15]; // powoduje utworzenie piętnastoelementowej
```

```
    // tablicy typu float. Tablica nie ma
```

```
    // nazwy, ale wskaźnik jest informowany o
```

```
    // jej adresie
```

```
delete [] tab; // powoduje skasowanie tablicy
```

- Przykład:

```
int rozmiar;
```

```
scanf( "%d" , &rozmiar );
```

```
long *Tablica = new long [ rozmiar ];
```

## Tablice wielowymiarowe

- Wczytywanie macierzy wierszami:

```
const int Wie = 10, Kol = 5;
```

```
int MM [ Wie ][ Kol ];
```

```
for (int i = 0; i < Wie; ++i)
```

```
    for (int j = 0; j < Kol; ++j)
```

```
        scanf("%d", &MM[ i ][ j ]);
```

```
// to samo co powyżej, ale z wykorzystaniem wskaźników
```

```
int ile = Wie * Kol, k;
```

```
int *p = &MM[0][0];
```

```
k = 0;
```

```
// tablica dwuwymiarowa to w rzeczywistości tablica jednowymiarowa
```

```
while (k++ < ile)
```

```
    scanf("%d", p++);
```