

Podsumowanie mini projektów z tematu wskaźników:

Szczypta statystyki:

Poszło Państwu całkiem nieźle, średnio wykonaliście Państwo polecenia od 1- 5/6 na 7 zadanych. Gratulacje!



W odniesieniu do zadań - generalizując:



1. Nie mam uwag.
2. Nie mam uwag.
3. Części z Państwa sprawił problem drugi ze sposobów ustawienia wskaźnika na adres zerowy.

I. Sposób:
`int *wsk1;`
`wsk1 = nullptr;`

lub w jednej operacji:

`int *wsk1 = nullptr;`

II. Sposób:

`int **wsk2{ }; // na sposób z klamrą trzeba inicjalizować od razu przy definiowaniu wskaźnika`



Zasadność stosowania adresu zerowego, w postaci `nullptr` dostrzeglibyście Państwo mając np. dwie funkcje:

```
void fun1 (int);  
void fun2 (int *);
```

Przy próbie ustawienia wskaźnika (z `fun2`) na adres zerowy za pomocą `nullptr` nie ma wątpliwości, która z funkcji zostanie wywołana.

Natomiast jeśli zastosujemy znany Państwu z języka C `NULL`, czyli (typ `int`), wówczas wywołania zostanie funkcja `fun1`, a nie o to nam chodziło.

4. Nie mam uwag.
5. Nie mam uwag.
6. Nie mam uwag.

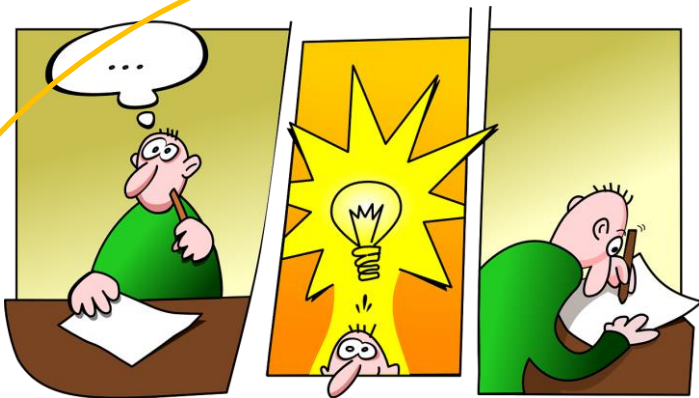
7. Do obsługi błędu, związanego z dynamiczną alokacją tablicy, w bloku `try-catch` można było posłużyć się nowym sposobem rzucania wyjątku - wprowadzoną przez standard C++11 - `bad_alloc`:



```
try  
{  
    int **wsk = new ...  
    //dalsze fragmenty kodu  
}  
  
catch (bad_alloc)  
{  
  
    cout << "Nie można przydzielić pamięci!" << endl;  
}
```

Alternatywnie (za pomocą instrukcji warunkowej if):

```
if (tablica != nullptr)
{
    cout << "Udana alokacja pamieci!" << endl;
    for (int i = 0; i < w; i++)
    {
        for (int j = 0; j < k; j++)
        {
            cout << "Wiersz " << i + 1 << " Kolumna " << j + 1 << " " <<
tablica[i][j] << endl;
        }
    }
}
else
    cout << "Nie można przydzielić pamieci!" << endl;
```



Pamiętajcie też Państwo, że język C++ wprowadził operatory:

- **new** (w miejsce znanej z języka C funkcji malloc)
- **delete** (w miejsce funkcji free)

Przykładowo:

- Tworzenie nowej dwuwymiarowej tablicy za pomocą operatora new:

```
int **tablica = new int *[w];
```

- Zwalnianie pamięci dla przydzielonej tablicy:
`delete[] tablica;`