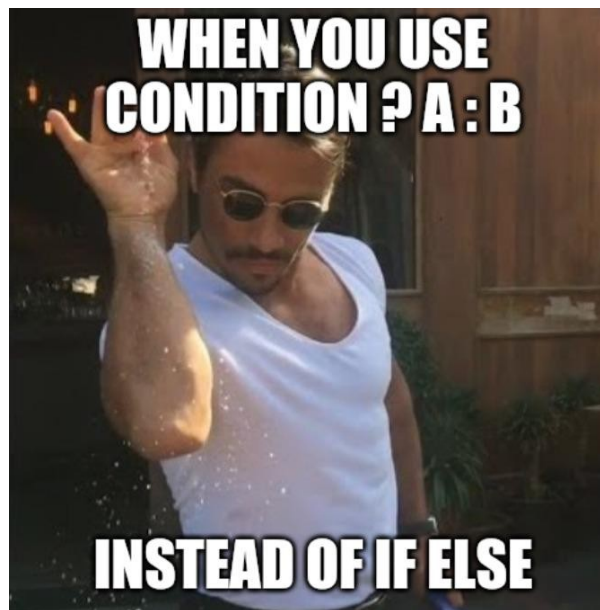


Семинар 3 по УП

Switch Case. Тернарнен оператор. Масиви



Switch / Case

- Защото като ще е гадно, поне да е по-малко гадно.
- Когато възможностите за едно нещо са много. Пример: по колко различни начина можеш да не завършиш ФМИ. Очевидно по повече, отколкото можеш.

Малко код

```
switch(a)
{
    case 1:
    {
        //Какво да се изпълни за a = 1
        break;
    }
    case 2:
    {
        //Какво да се изпълни за a = 2
        break;
    }
    case 3:
    {
        //Какво да се изпълни за a = 3
        break;
    }
    default:
    {
        //Ако никъде не влезе
        break;
    }
}
```

Сега какво забелязваме

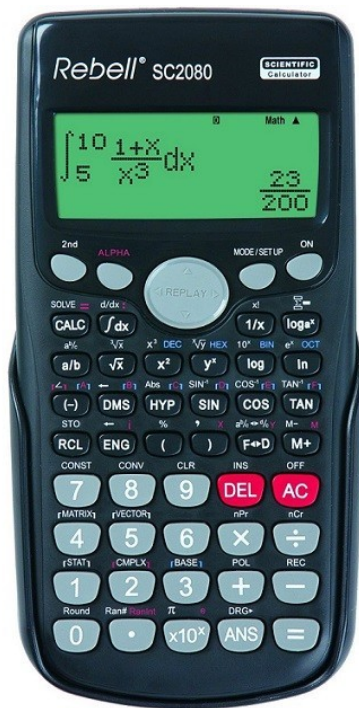
- Разглеждаме случаи за a . Малко като в математиката
- Ако не влезем в никой случай отиваме в default.
- Защо default е най-отдолу?

Пример

```
int main()
{
    int option;
    std::cout << "Избери име: " << std::endl
               << "1) Иван"      << std::endl
               << "2) Гошо"      << std::endl
               << "3) Пешо"      << std::endl;
    switch(option)
    {
        case 1:
        {
            std::cout << "Ванката е топ пичага" << std::endl;
            break;
        }
        case 2:
        {
            std::cout << "Гошо е кварталния пияница" << std::endl;
            break;
        }
        case 3:
        {
            std::cout << "Пешо така и няма да си вземе изпита по УП" << std::endl;
            break;
        }
        default:
        {
            std::cout << "Невалидно име!" << std::endl;
            break;
        }
    }
}
```

Задача

- Направете прост калкулатор със Switch.



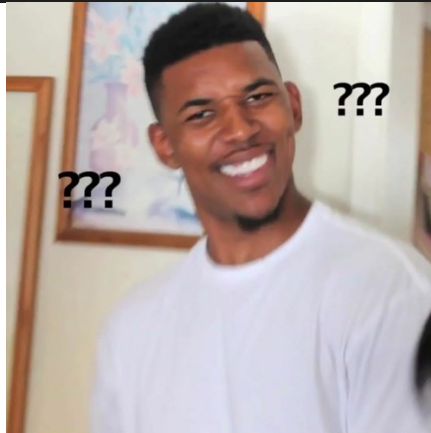
- Елката на баба, която за разлика от мен може да решава интеграли.

Тернарен оператор

- Достигаме ново ниво на мързел.
- В основата на всичко лежи фактът че за да напишем if, трябва да напишем цели 2 букви и за противния случай – още 4..., което е много работа. (Игнорирайте факта че изписвам всичко това на тази презентация, а да пиша if ме мързи)

Как работи това нещо

```
int main()  
{  
    int a;  
    std::cin >> a;  
    std::cout << ((a % 2 == 0) ? "Yes" : "No") << std::endl;  
}
```



Задачи

- Изведете дали година е високосна с тернарен оператор
- След тежката нощ, Мишо излиза от дискотеката и поглежда такситата, но хубавите момичета вече са ги взели. Мишо е принуден да отвори такси приложението. То му предлага 3 таксита. Вашата задача е по подадени 3 цени на такситата (цели числа), да ги подредите по големина (от най-евтиното до най-скъпото).
- Дадена е променлива a . Да и се присвои стойност $a/2$, ако е четно числото в нея и $a * 2$ – ако е нечетно.

Масиви

- Защото искаме от всичко по много
- Масивите са просто много променливи от един тип, които са представени последователно в паметта.
- Отделните променливи се наричат елементи на масива, а номерът им – индекс. Индексацията започва от 0.

Чакай, чакай какво? Объркахме се!!!

Memory Location									
200	201	202	203	204	205	206			
U	B	F	D	A	E	C			
0	1	2	3	4	5	6			
Index									

Index

Достъп до елементите

- $a[index]$, където на мястото на $index$ слагаме номера на елемента, който искаме да достъпим
- Въпрос за.... 0 лева (и аз съм беден студент като вас): Как ще въведем масив от клавиатурата?

Да с цикъл! Пример:

```
int arr[10];  
for(int i = 0; i < 10; i++)  
{  
    std::cin >> arr[i];  
}
```

Понеже всичко е ясно. Въпроси:

- Как ще изведем масива?
- Как ще вземем 4тия елемент на масива от примера?
- Как ще вземем 15тия елемент на масива от примера?

Задачи

- Въведете n елемента в масив и ги обърнете в друг
- Въведете едно число в двоична система с максимум 100 символа и го изведете в десетична
- Въведете n числа и изведете само четните от тях
- Мишо има n лева. В дискотеката се предлагат m различни питиета, на различни цени. Намерете най-скъпото, което Мишо може да си позволи.
- Да се напише програма, която получава цяло неотрицателно число n и масив от n цели числа. Програмата да проверява дали въведените числа образуват редица „трион“. Редица „трион“ наричаме последователност, за която е вярно, че $a_1 \leq a_2 \geq a_3 \leq a_4 \dots$ или $a_1 \geq a_2 \leq a_3 \geq a_4 \dots$ (Леко променена 2 зад. от контролно 2 по УП вариант 1.)
- От стандартния вход се въвежда естественото число n , следвано от други $2 \times n$ на брой дробни числа (представени чрез тип `double`) $x_0, y_0, x_1, y_1, \dots, x_{n-1}, y_{n-1}$. Двойките (x_i, y_i) са декартови координати на точки в евклидовата равнина. Да се намери правоъгълник с възможно най-малка площ, чиито страни са успоредни на координатните оси и който съдържа всички въведени точки. Да се изведат на стандартния изход координатите на горния ляв ъгъл на правоъгълника и дължините на страните му.
Пример: За редицата от точки $(0, 0), (-1, 0), (2, 1)$ програмата ще изведе -1 1 3 1. (Задача 3 от контролно 2 миналата година)