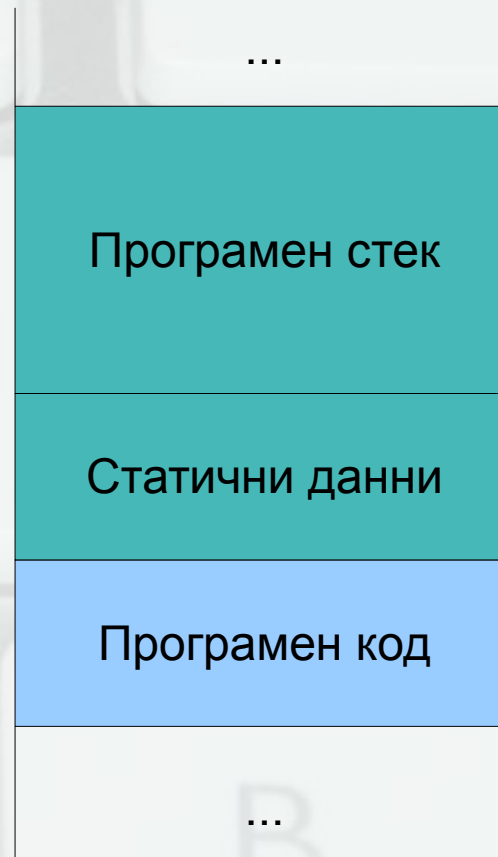


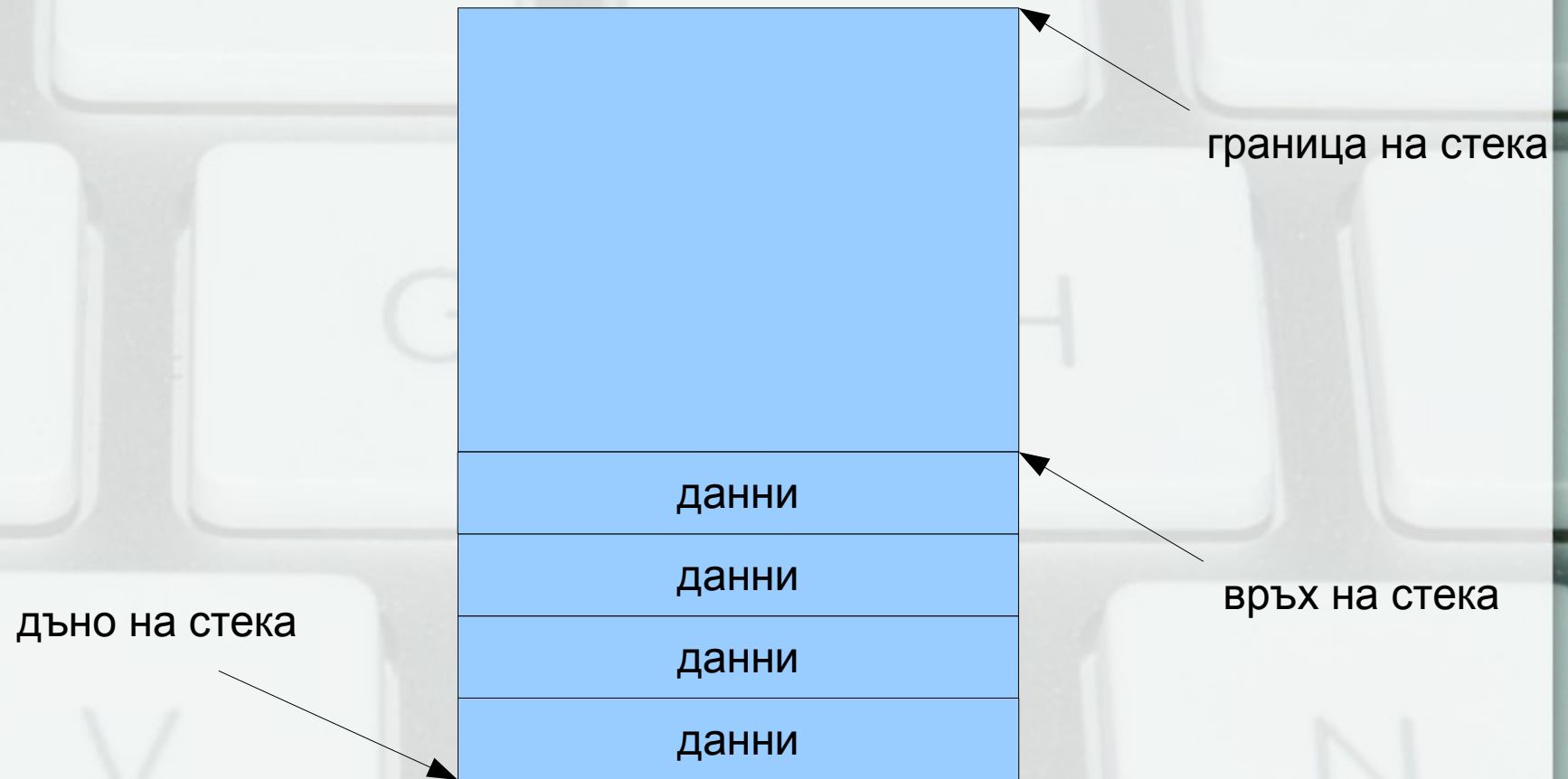
Функции

(часть 3)

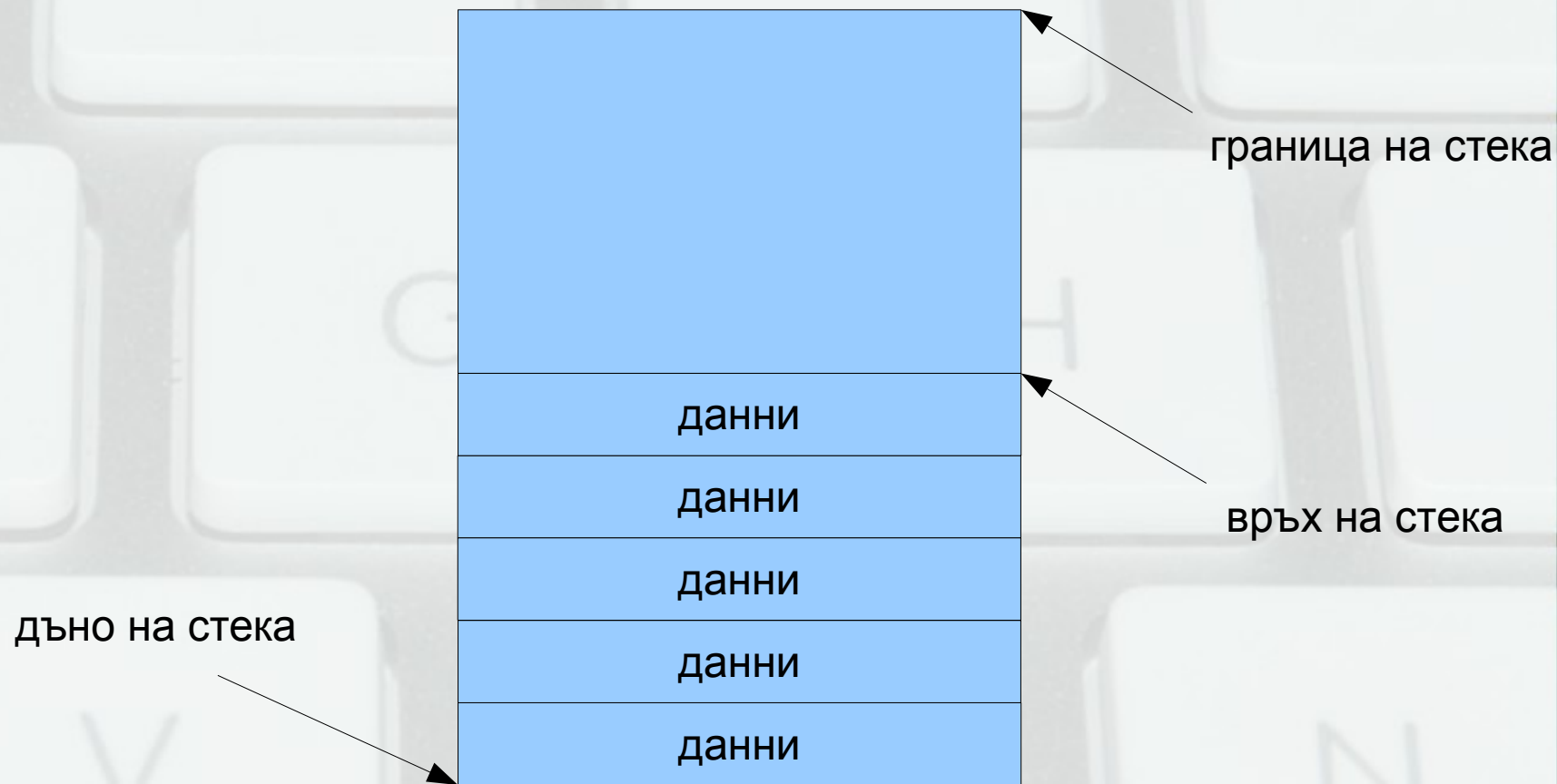
Схема на програмната памет



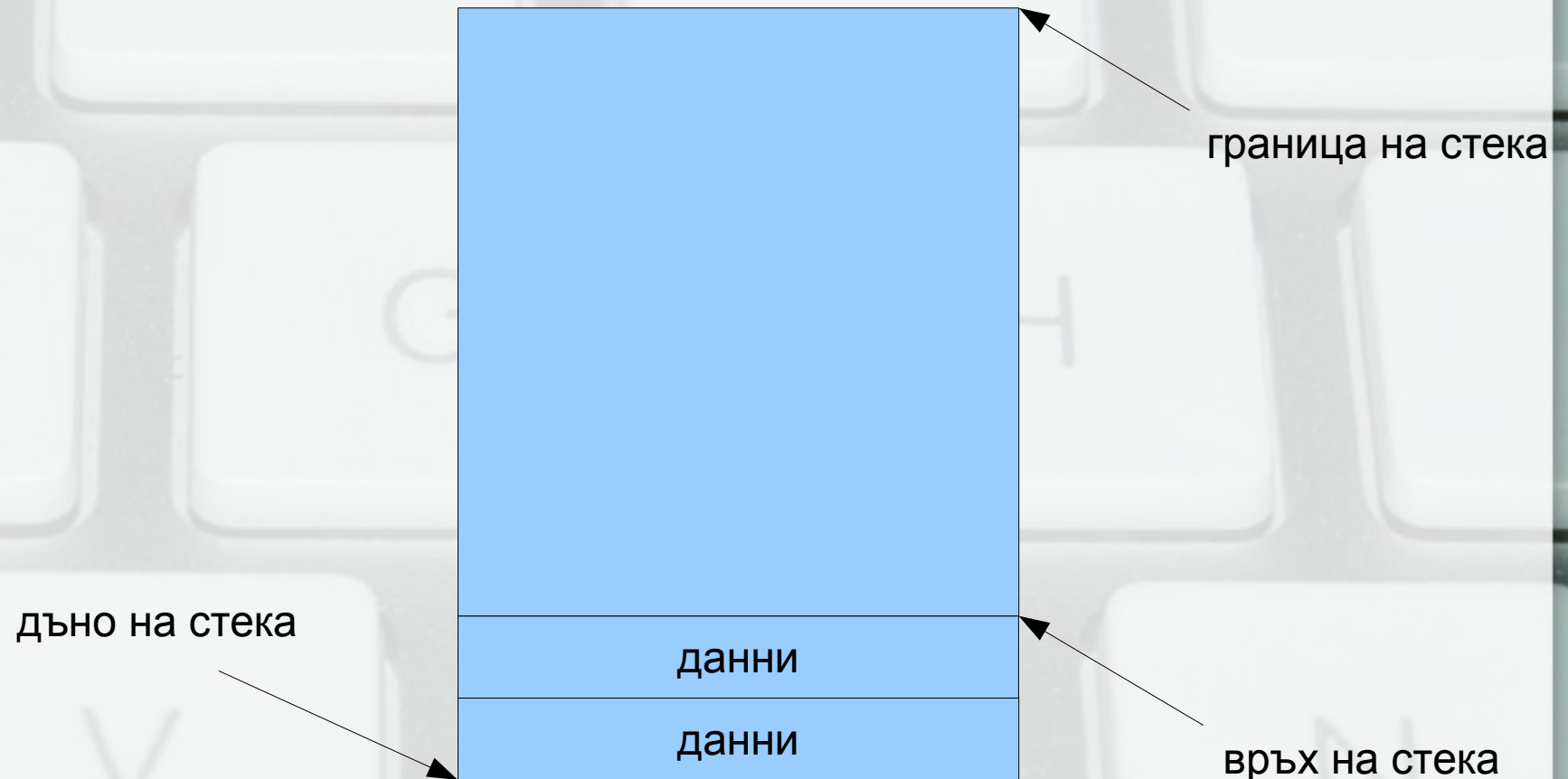
Програмен стек



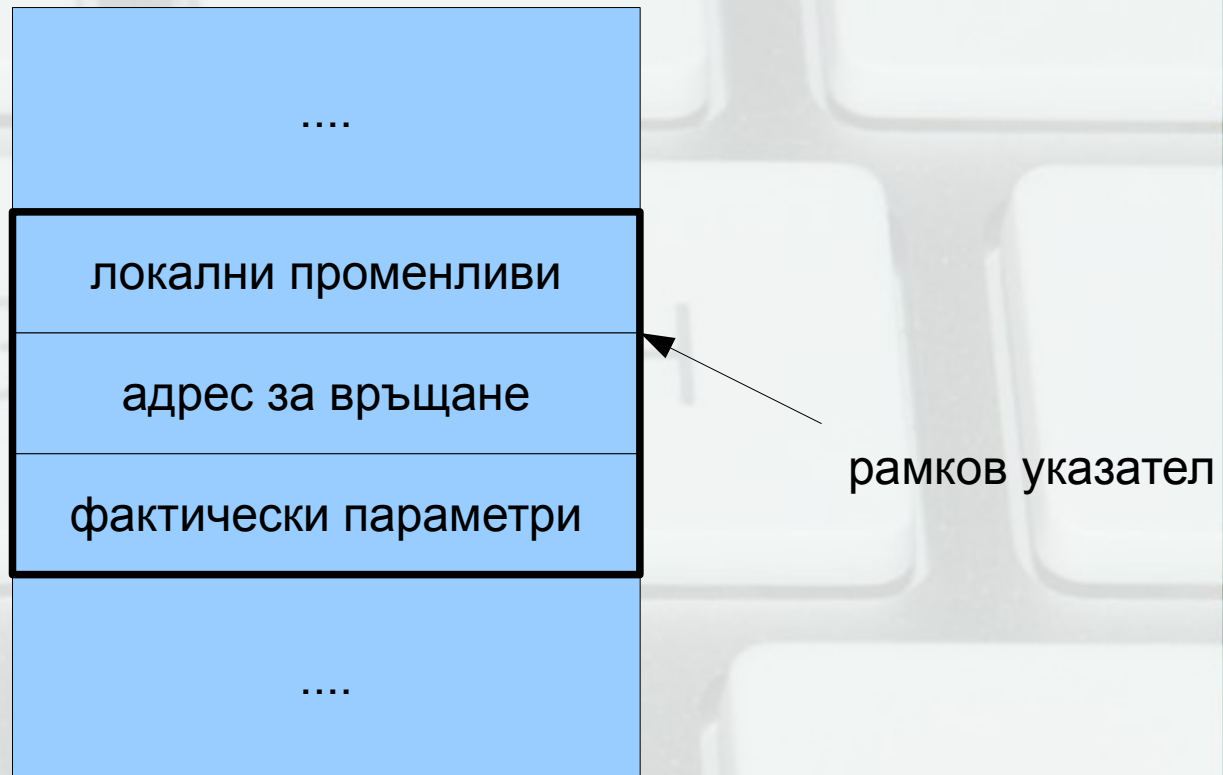
Програмен стек



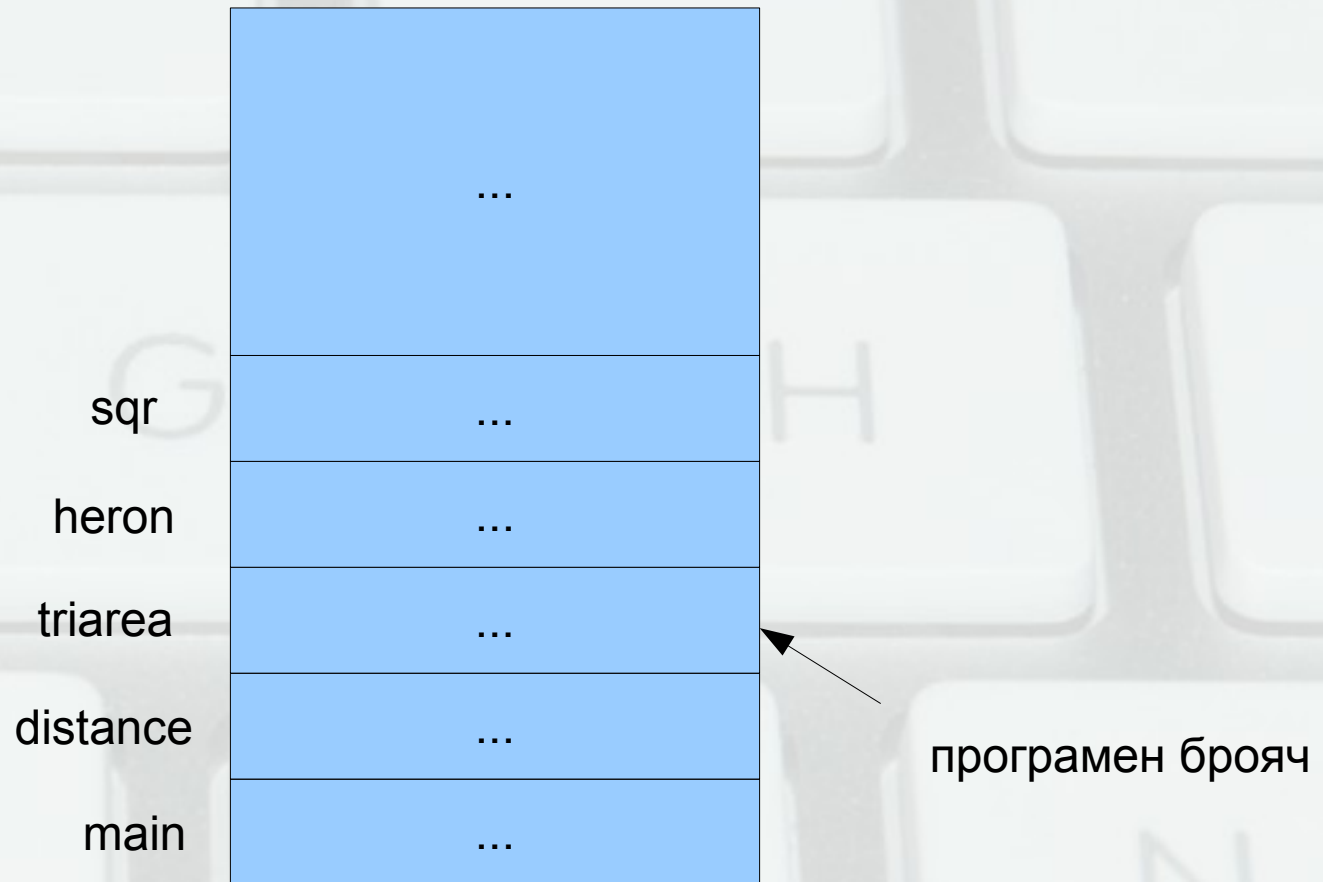
Програмен стек



Стекова рамка на функция



Област за програмен код



Предаване по стойност (call by value)

- пресмята се стойността на фактическия параметър
- в стековата рамка на функцията се създава копие на стойността
- всяка промяна на стойността остава локална за функцията
- при завършване на функцията, предадената стойност и всички нейни промени изчезват

Предаване по указател (адрес) (call by pointer)

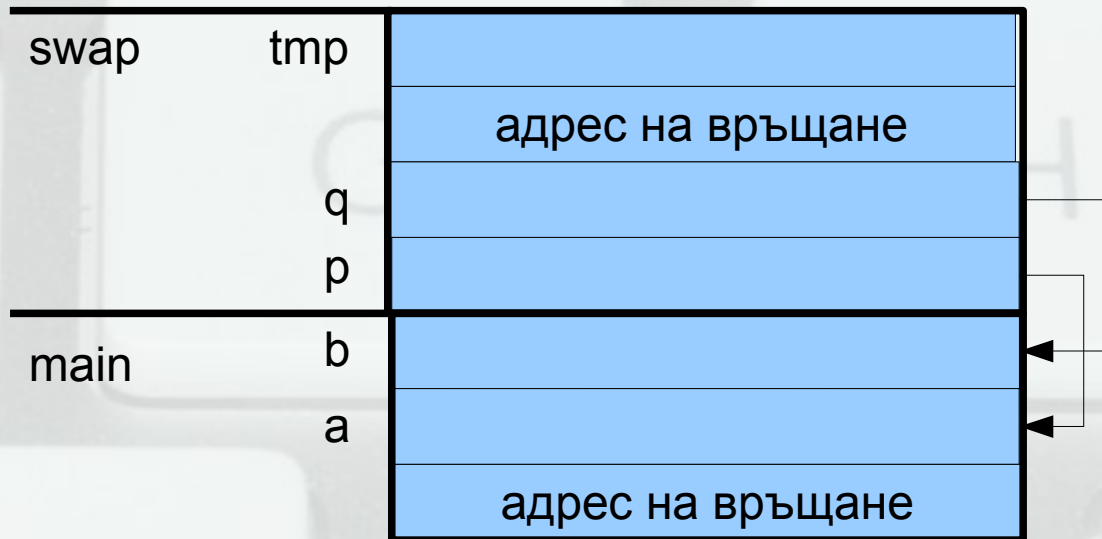
- Предава се **адрес**, вместо стойност
- Фактическите параметри трябва да са от тип “указател към нещо”
- Функцията може през указателите да променя стойности на външни променливи

Предаване по указател (адрес) (call by pointer)

- Пример: Размяна на две променливи

```
void swap(int* p, int* q) {  
    int tmp = *p; *p = *q; *q = tmp;  
}  
int main() {  
    int a = 5, b = 8;  
    swap(&a, &b);  
    cout << a << ' ' << b << endl;  
}
```

Стекова рамка при предаване по указател

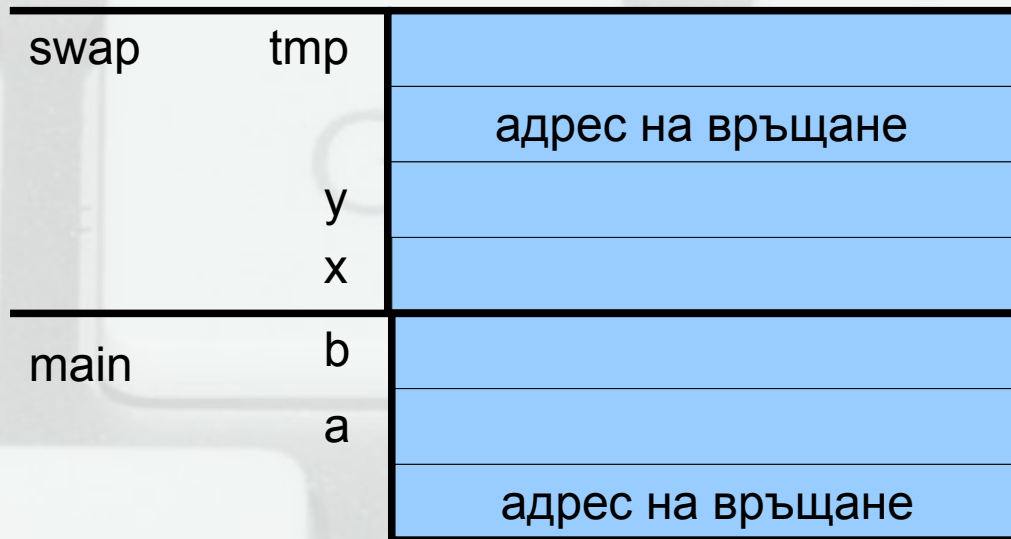


Предаване по псевдоним (референция) (call by reference)

- Пример: Размяна на две променливи

```
void swap(int& x, int& y) {  
    int tmp = x; x = y; y = tmp;  
}  
int main() {  
    int a = 5, b = 8;  
    swap(a, b);  
    cout << a << ' ' << b << endl;  
}
```

Стекова рамка при предаване по псевдоним



Предаване на масиви като параметри

- `<параметър_масив> ::=`
 `<тип> <име> [[<константен_израз>]] |`
 `<тип>* <име>`
- всъщност масивите се предават по указател
- ...затова изразът в скобите се игнорира!
- ...затова промените в масива винаги се отразяват в оригинала

Предаване на многомерни масиви като параметри

- `<параметър_многомерен_масив> ::=`
 `<тип> <идентификатор> [[<константа>]]`
 `{[<константа>]}` |
 `<тип> (*<идентификатор>) {[<константа>]}`
- **Внимание!** `int* a[10]` е различно от `int (*a)[10]`!
- константата в първите скоби се игнорира!
- многомерните масиви също се предават по указател

Предаване на многомерни масиви като параметри

- $\langle \text{параметър_многомерен_масив} \rangle ::=$
 $\langle \text{тип} \rangle \langle \text{идентификатор} \rangle \{ \langle \text{константа} \rangle \}$
 $\{ \langle \text{константа} \rangle \} |$
 $\langle \text{тип} \rangle (* \langle \text{идентификатор} \rangle) \{ \langle \text{константа} \rangle \}$
- първата размерност трябва да се подаде като допълнителен параметър
- другите размерности се предават, за да се пресмятат правилно позициите на елементите

Примерни функции

- Извеждане на матрица от числа
- Прочитане на масив от низове
- Проверка за срещане на дума в масив от низове
- Умножение на две правоъгълни матрици

Указателите като върнат резултат

- **Внимание:** хубаво е да се връщат указатели към обекти, които ще продължат да съществуват след като функцията приключи

```
int* pointMax(int* p, int* q) {  
    if (*p > *q)  
        return p;  
    return q;  
}  
int* r = pointMax(&a, &b); (*r)--;
```


Псевдонимите като върнат резултат

- Важи същото правило като за указателите

- ```
int& middle(int& x, int& y, int& z) {
 if (x <= y && y <= z || z <= y && y <= x)
 return y;
 if (y <= z && z <= x || x <= z && z <= y)
 return z;
 return x;
}
```

- `middle(x, y, z) = 5;`

# Масивите като върнат резултат

- Функциите не могат да имат за тип на връщан резултат масив
- Но могат да връщат тип указател
- По този начин може да се връщат едномерни масиви
- **Внимание:** връщат се само масиви, които ще продължат да съществуват след като функцията завърши

# Примери

- Връщане на позицията на първото срещане на даден символ в низ
- Връщане на позицията на първото различие между два низа