



## Тест 5 - Сложни типове и структури от данни - група 1

Имейл адресът на отговарящия (aleksandar.m.yosifov.2022@elsys-bg.org) бе записан при изпращането на този формуляр.

Удостоверяване

Парола \*

nloUsER

### Въпроси

Какво ще изведе следната програма: \*

```
#include <stdio.h>
```

```
typedef struct {
    char name[30];
    int number;
} Student;
```

```
typedef enum {
    SOFTWARE = 1,
    NETWORKS,
} Specialty;
```

```
typedef struct {
    Student students[2];
    struct {
        char firstname[30];
        char surname[30];
    } homeroomteacher;
    Specialty specialty;
} StudentClass;
```

```
int main(void) {
```

```
    StudentClass sclass = {
        students: {
            [1] = {"Maria", 2},
            [0] = {"Nikola", 1}
        },
        homeroomteacher = {
            firstname: "Lybomir",
            surname: "Dimitrov"
        },
        specialty: SOFTWARE
    };
```

```
    printf(
        "%s%s%d",
        sclass.students[0].name,
        sclass.homeroomteacher.surname,
        sclass.specialty
    );
```

```
    return 0;
```

```
}
```

NikolaDimitrovSOFTWARE

Какво декларира следния фрагмент от код: \*

```
enum Bool {
    True = 1,
    False = 0
};
```

☒ Нов изброим тип

☐ Нова структура

☐ Ново обединение

☐ Това не е валидна конструкция в C

Какво декларира следния фрагмент от код: \*

```
union Node {
    long egn;
    char phone[12];
};
```

☐ Нов изброим тип

☐ Нова структура

☒ Ново обединение

☐ Това не е валидна конструкция в C

Какво е характерно за едносвързания списък? (Изберете всичко, което се отнася за него) \*

☒ Структурата е динамична (може да променя размера си)

☐ Структурата е статична (не може да променя размера си)

☐ Достъпът до елементите по индекс е бърза операция

☒ Достъпът до елементите по индекс е бавна операция

☒ Изтриването и добавянето на елементи в началото е бърза операция

☐ Изтриването и добавянето на елементи в началото е бавна операция

☒ Използва рекурсивна структура в имплементацията си

☒ Държи елементите във вътрешен масив

Какво декларира следния фрагмент от код: \*

```
typedef StudentClass {
    char classname;
    int grade;
};
```

☐ Нов изброим тип

☐ Нова структура

☐ Ново обединение

☒ Това не е валидна конструкция в C

Какво е предназначението на ключовата дума "typedef"? \*

☐ Декларира структура

☒ Дава възможност да се задават нови имена на съществуващи типове

☐ Дефинира анонимен тип

☐ Декларира константен тип

Кой/кои са правилните начини да се достъпи член променливата "name", през променливата "ivan" в следната програма: \*

```
#include <stdio.h>
```

```
typedef struct {
    char name[30];
    int number;
} Student;
```

```
int main(void) {
```

```
    Student * ivan;
```

```
    return 0;
```

```
}
```

☐ ivan.name

☒ ivan->name

☐ (\*ivan).name

☒ (&ivan).name

Какво представлява анонимния тип? \*

☐ Това е тип, който може да се използва вместо функции

☒ Това е декларация на нов тип, при която се пропуска името на типа

☐ Това е тип, който съдържа променлива от самия себе си

☐ Това е тип, който задължително се дефинира с помощта на ключовата дума "typedef"

Какво ще изведе следната програма: \*

```
#include <stdio.h>
```

```
int main(void) {
```

```
    int arr[7] = {
```

```
        [3] = 2,
```

```
        [0]3,
```

```
        [4 ... 6] = 5
```

```
    };
```

```
    for (int i = 0; i < 7; i++) {
```

```
        printf("%d", arr[i]);
```

```
    }
```

```
    return 0;
```

```
}
```

0002555

Какво е предназначението на следната функция? : \*

```
void foo(LinkedList *list, int index, double value) {
    if (index == 0) {
        pushfront(list, value);
    } else {
        struct ListNode * prevnode = get(list, index - 1);
        struct ListNode * newnode = createnode(value);
        newnode->next = prevnode->next;
        prevnode->next = newnode;
        list->size++;
    }
}
```

☐ Да добави елемент в началото на едносвързан списък

☒ Да добави елемент на произволна позиция в едносвързан списък

☐ Да премахне елемент от началото на едносвързан списък

☐ Да премахне елемент от произволна позиция в едносвързан списък

Какво е характерно за динамичния масив? (Изберете всичко, което се отнася за него) \*

☒ Структурата е динамична (може да променя размера си)

☐ Структурата е статична (не може да променя размера си)

☒ Достъпът до елементите по индекс е бърза операция

☐ Достъпът до елементите по индекс е бавна операция

☒ Изтриването и добавянето на елементи в началото е бърза операция

☐ Изтриването и добавянето на елементи в началото е бавна операция

☐ Използва рекурсивна структура в имплементацията си

☒ Държи елементите във вътрешен масив

Какво е предназначението на следната функция? \*

```
int foo(DynamicArray * arr) {
    int newsize = arr->size - 1;
    arr->buffer[arr->size];
    arr->result[arr->size] = 0;
    resize(arr, newsize);

    return result;
}
```

☐ Да добави елемент в края на динамичен масив

☐ Да добави елемент на произволна позиция в динамичен масив

☒ Да премахне елемент от края на динамичен масив

☐ Да премахне елемент от произволна позиция в динамичен масив

Какво представлява рекурсивния тип? \*

☐ Това е тип, който може да се използва вместо функции

☐ Това е декларация на нов тип, при която се пропуска името на типа

☒ Това е тип, който съдържа променлива от самия себе си

☐ Това е тип, който задължително се дефинира с помощта на ключовата дума "typedef"

Какво декларира следния фрагмент от код: \*

```
struct Student {
    char name[30];
    int number;
    int age;
};
```

☐ Нов изброим тип

☒ Нова структура

☐ Ново обединение

☐ Това не е валидна конструкция в C

Какво ще изведе следната програма: \*

```
#include <stdio.h>
```

```
typedef struct {
    char name[30];
    int number;
} Student;
```

```
typedef enum {
    SOFTWARE = 1,
    NETWORKS,
} Specialty;
```

```
typedef struct {
    Student students[2];
    struct {
        char firstname[30];
        char surname[30];
    } homeroomteacher;
    Specialty specialty;
} StudentClass;
```

```
int main(void) {
```

```
    StudentClass sclass = {
        ({ivan, 1}, {Martin, 2}),
        ({Dragon, Ivanov},
        NETWORKS
    );
```

```
    printf(
        "%s%s%d",
        sclass.students[1].name,
        sclass.homeroomteacher.firstname,
        sclass.specialty
    );
```

```
    return 0;
```

```
}
```

MartinIvanovNETWORKS