

შესავალი დაპროგრამებაში

ლექტორი:

სალომე ონიანი

ტელ. 571 39 40 22

ელ.-ფოსტა salome.oniani@iliauni.edu.ge



მასივი

„მასივის ტიპი“ „მასივის სახელი“ [„მასივის სიგრძე“];

char symbols [256];

```
#include <stdio.h>
int main()
{
    int arr[10];
    printf("Please enter 10 munbers \n ");
    for (int i = 0; i < 10; i++)
    {
        printf("enter number %d - ", i+1);
        scanf("%d", &arr[i]);
    }
    printf("So, you entered this array: ");
    for (int i = 0; i < 10; i++) printf("%d ", arr[i]);
    int doubled = 0;
    for(int i = 0; i<10-doubled; i++)
        for (int j=i+1; j<10 -doubled; j++)
            if (arr[i] == arr[j]){
                for(int k=j; k<9-doubled; k++) arr[k]=arr[k+1];
                doubled++;
            }
    printf("\nAfter delete duplicate elements we get array: ");
    for (int i = 0; i < 10-doubled; i++) printf("%d ", arr[i]);
    return 0;
}
```



დაწერეთ პროგრამა, სადაც მომხმარებელი შეძლებს
10 რიცხვის შეტანას და პროგრამა დაუბეჭდავს
ყველაზე ხშირად განმეორებად რიცხვს

```
#include <stdio.h>
#include <string.h>
int main()
{
    int arr[10];
    printf("Please enter 10 munbers \n ");
    for (int i = 0; i < 10; i++)
    {
        printf("enter number %d - ", i+1);
        scanf("%d", &arr[i]);
    }
    int max=arr[0];
    for(int i=1; i<10; i++){
        if(max<arr[i]) max = arr[i];
    }
    int doubled [max+1];
    memset(doubled, 0, sizeof(doubled));
    for(int i=0; i<10; i++) doubled[arr[i]]++;
    int m = doubled[0];
    int index = 0;
    for(int i=1; i<=max; i++){
        if(m<doubled[i]) {
            m = doubled[i];
            index = i;
        }
    }
    printf("\nthe most repeated number is %d", index);

    return 0;
}
```



სასწავლო კვირა VIII

- სტრუქტურები.
- სტრუქტურა, როგორც მონაცემთა ახალი ტიპის შემოღების შესაძლებლობა.
- სტრუქტურის ველები.
- სტრუქტურების შედარება მასივებთან.
- ადგილის გამოყოფა სტრუქტურისათვის.
- სტრუქტურის ინიციალიზაცია.
- გაერთიანებები.
- გაერთიანებების შედარება სტრუქტურასთან.
- ტიპისათვის ახალი სახელის დარქმევა typedef კონსტრუქციის გამოყენებით.

```
memset(array, 0, n);
```

სჭირდება `#include <string.h>` ბიბლიოთეკა

ეს არის ფუნქცია, რომელიც გამოიყენება მასივის პირველ `n` ელემენტში რაიმე მონაცემის ჩასაწერად

მაგალითად: `memset(array, 2, 10);`

ეს ჩანაწერი `array` მასივის პირველ 10 უჯრაში ჩაწერს 2-იანს

```
sizeof(array));
```

აქ `sizeof` ფუნქცია აბრუნებს მასივის ელემენტების რაოდენობას

```
memset(doubled, 0, sizeof(doubled));
```

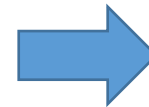
ამ ჩანაწერით ხდება `doubled` მასივის 0-ებით შევსება

სტრუქტურები

მონაცემთა ბაზა, რომელშიც შესაძლებელია სხვადასხვა ტიპის მონაცემების შენახვა

```
struct „სტრუქტურის სახელი“  
{  
    სტრუქტურის წევრები;  
};
```

სტრუქტური წევრები შეიძლება იყოს განსხვავებული ტიპის ცვლადები და მათ შეგიძლიათ დაარქვათ თქვენთვის სასრუველი სახელები



char
int
float
double

სტრუქტურები

მაგალითად, თუ გვჭირდება დავწეროთ პროგრამა რომელიც მოგვცემს საშუალებას შევინახოთ სტუდენტის მონაცემები, ამისათვის გვჭირდება:

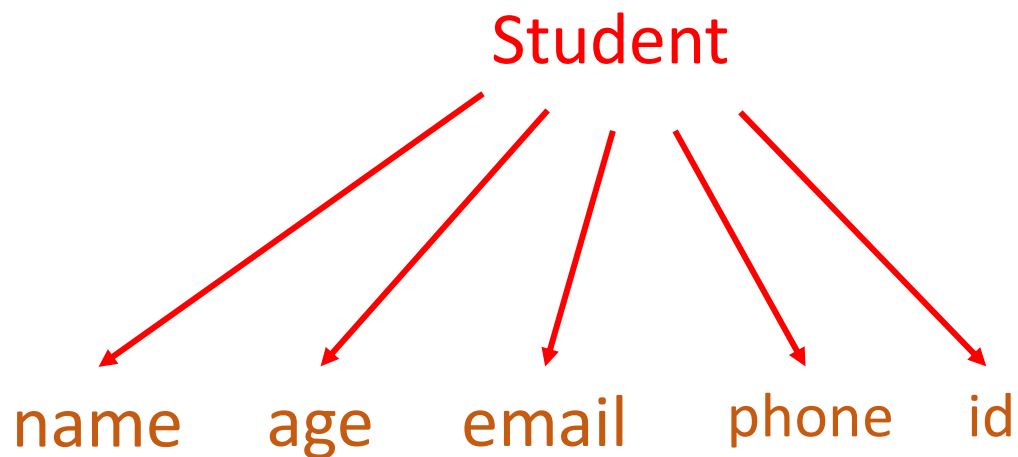
- სიმბოლოების მასივი (სტუდენტის სახელისათვის) → `char name[50];`
- მთელი ტიპის ცვლადი (სტუდენტის ასაკისათვის) → `int age;`
- სიმბოლოების მასივი (სტუდენტის მეილისათვის) → `char email[30];`
- მთელი ტიპის ცვლადი (სტუდენტის ტელეფონის ნომრისათვის) → `unsigned long int phone;`
- მთელი ტიპის ცვლადი (სტუდენტის პირადი ნომერისათვის) → `double id;`

დადებითი მთელი
ტიპის მონაცემი,
რომელსაც შეუძლია 0-
დან 4,294,967,295
მნიშვნელობის შენახვა

სტრუქტურები

```
struct Student{  
    char name[50];  
    int age;  
    char email[30];  
    unsigned long int phone;  
    double id;  
};
```

სტუდენტის სტრუქტურის შექმნა

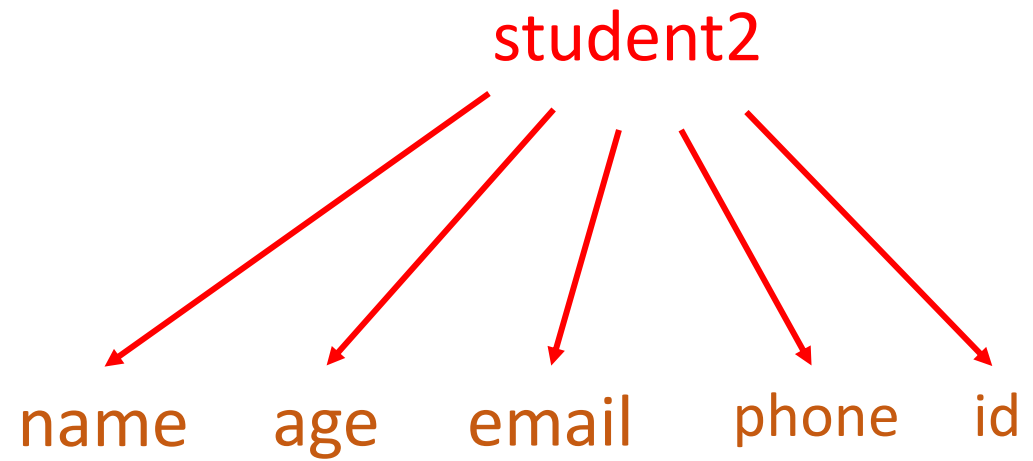
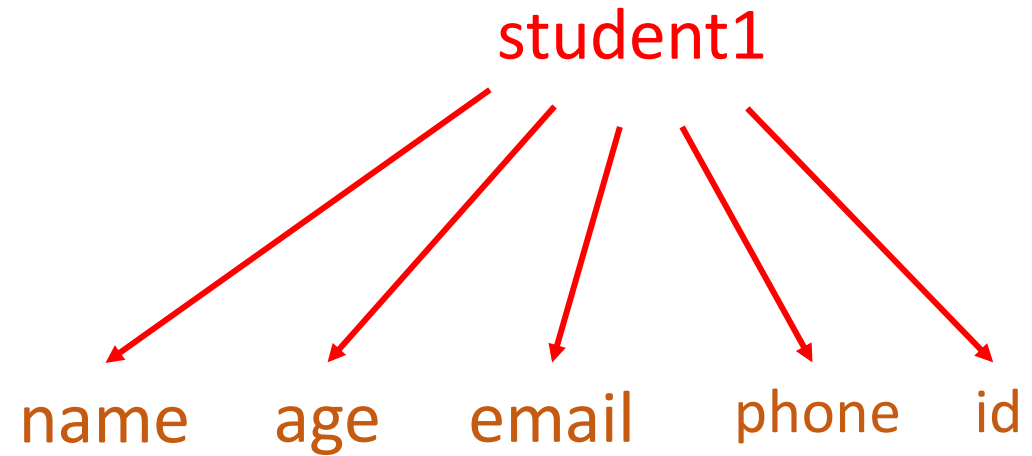


სტრუქტურები

სტუდენტის სტრუქტურისათვის ცვლადის განსაზღვრა

```
struct Student{  
    char name[50];  
    int age;  
    char email[30];  
    unsigned long int phone;  
    double id;  
} student1, student2;
```

student1 და student2 არის ცვლადის
სახელები, რომლებიც წარმოადგენენ
სტრუქტურის ტიპის ცვლადებს



სტრუქტურები

```
struct Student{  
    char name[50];  
    int age;  
    char email[30];  
    unsigned long int phone;  
    double id;  
};
```

```
struct Friends{  
    char name[50];  
    int age;  
    char email[30];  
    unsigned long int phone;  
    double id;  
};
```

სტრუქტურის წევრების სახელები შეიძლება გამოვრდეს სხვა სტრუქტურაში

არ შეიძლება ორ სტრუქტურას ერთნაირი სახელი ერქვას

სტრუქტურები

სტუდენტის სტრუქტურის ცვლადის ინიციალიზაცია

```
struct Student{  
    char name[50];  
    int age;  
    char email[30];  
    unsigned long int phone;  
    double id;
```

```
} student1 = {"salome", 35, "salome@gmail.com", 557848484, 12345678901},  
student2 = {"ana", 20, "a.dad@gmail.com", 895656565, 22345678901};
```

ასეთი ინიციალიზაციის დროს აუცილებელია
დავიცვათ ის თანმიმდევრობა რა
თანმიმდევრობითაც აღვწერეთ სტრუქტურის
წევრები

student1	name	salome
	age	35
	email	salome@gmail.com
	phone	557848484
	id	12345678901

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

```
struct Student{  
    char name[50];  
    int age;  
    char email[30];  
    unsigned long int phone;  
    double id;  
}  
student1 = {"salome",35,"salome@gmail.com",557848484,12345678901},  
student2 = {"ana",20,"a.dad@gmail.com",895656565,22345678901};  
  
main (){  
    printf("name %s\n", student1.name);  
    printf("email %s\n", student1.email);  
    printf("age %d\n", student1.age);  
    printf("id %.0f\n", student1.id);  
    printf("phone %d\n", student1.phone);  
  
    return 0;  
}
```

სტრუქტურის წევრებში შენახული მონაცემების
დაბეჭდვა

C:\Users\sалome\Documents\Untitled1.exe

```
name salome  
email salome@gmail.com  
age 35  
id 12345678901  
phone 557848484
```

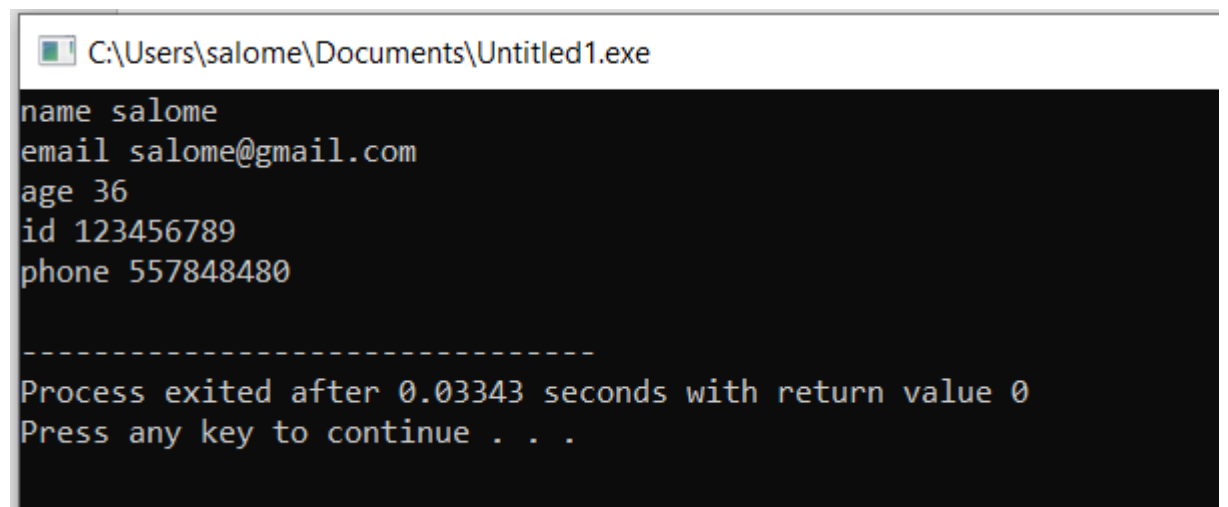
```
-----  
Process exited after 0.03331 seconds with return value 0  
Press any key to continue . . .
```

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

```
struct Student{  
    char name[50];  
    int age;  
    char email[30];  
    unsigned long int phone;  
    double id;  
}  
student1 = {"salome",35,"salome@gmail.com",557848484,12345678901},  
student2 = {"ana",20,"a.dad@gmail.com",895656565,22345678901};
```

```
main (){  
    printf("name %s\n",student1.name);  
    printf("email %s\n",student1.email);  
    student1.age++;  
    printf("age %d\n",student1.age);  
    student1.id /=100;  
    printf("id %0.f\n",student1.id);  
    student1.phone -=4;  
    printf("phone %d\n",student1.phone);  
    return 0;  
}
```

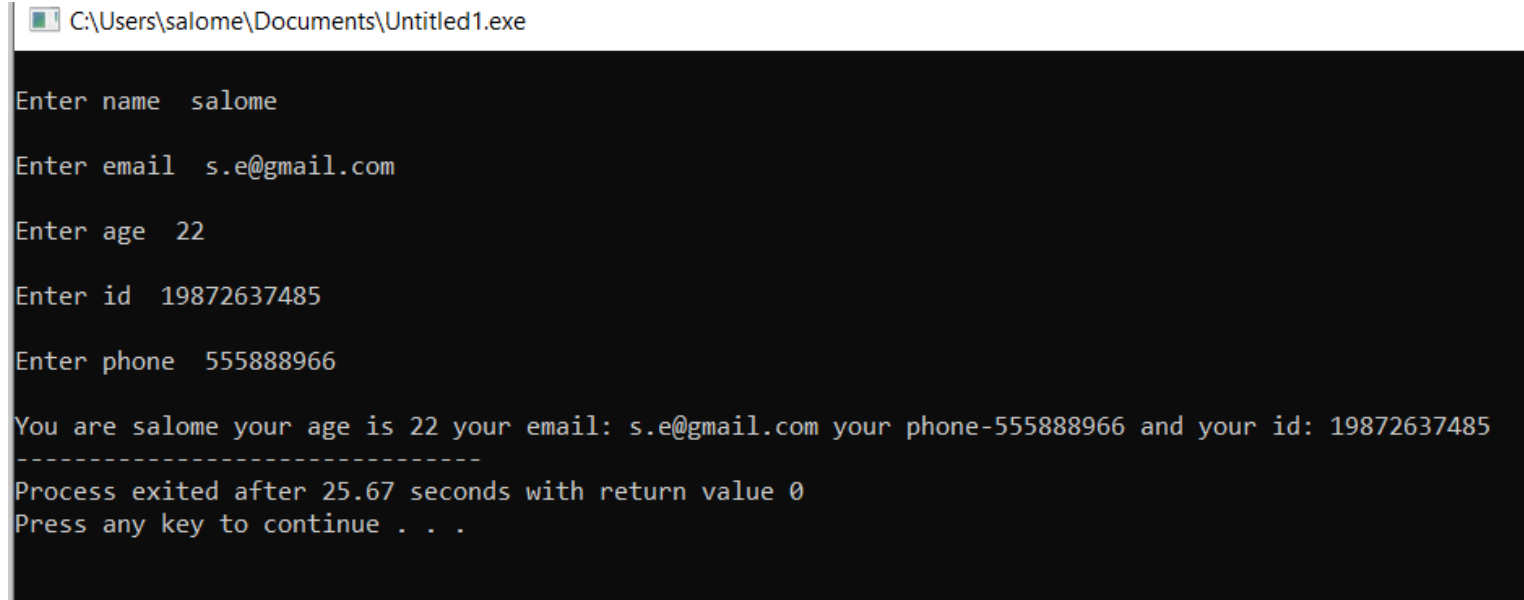
სტრუქტურის წევრებში შენახული მონაცემების
ცვლილება



```
C:\Users\salome\Documents\Untitled1.exe  
name salome  
email salome@gmail.com  
age 36  
id 123456789  
phone 557848480  
-----  
Process exited after 0.03343 seconds with return value 0  
Press any key to continue . . .
```

```
#include <stdio.h>
struct Student{
    char name[50];
    int age;
    char email[30];
    unsigned long int phone;
    double id;
}student1, student2 ;
main (){
    printf("\nEnter name ");
    scanf("%s",&student1.name);
    printf("\nEnter email ");
    scanf("%s",&student1.email);
    printf("\nEnter age ");
    scanf("%d",&student1.age);
    printf("\nEnter id ");
    scanf("%lf",&student1.id);
    printf("\nEnter phone ");
    scanf("%d",&student1.phone);
    printf("\nYou are %s your age is %d your email: %s your phone-%d and your id: %0.f", student1.name,
        student1.age, student1.email, student1.phone, student1.id);
    return 0;
}
```

სტრუქტურის წევრებში კონსოლის ეკრანიდან
ინფორმაციის შენახვა

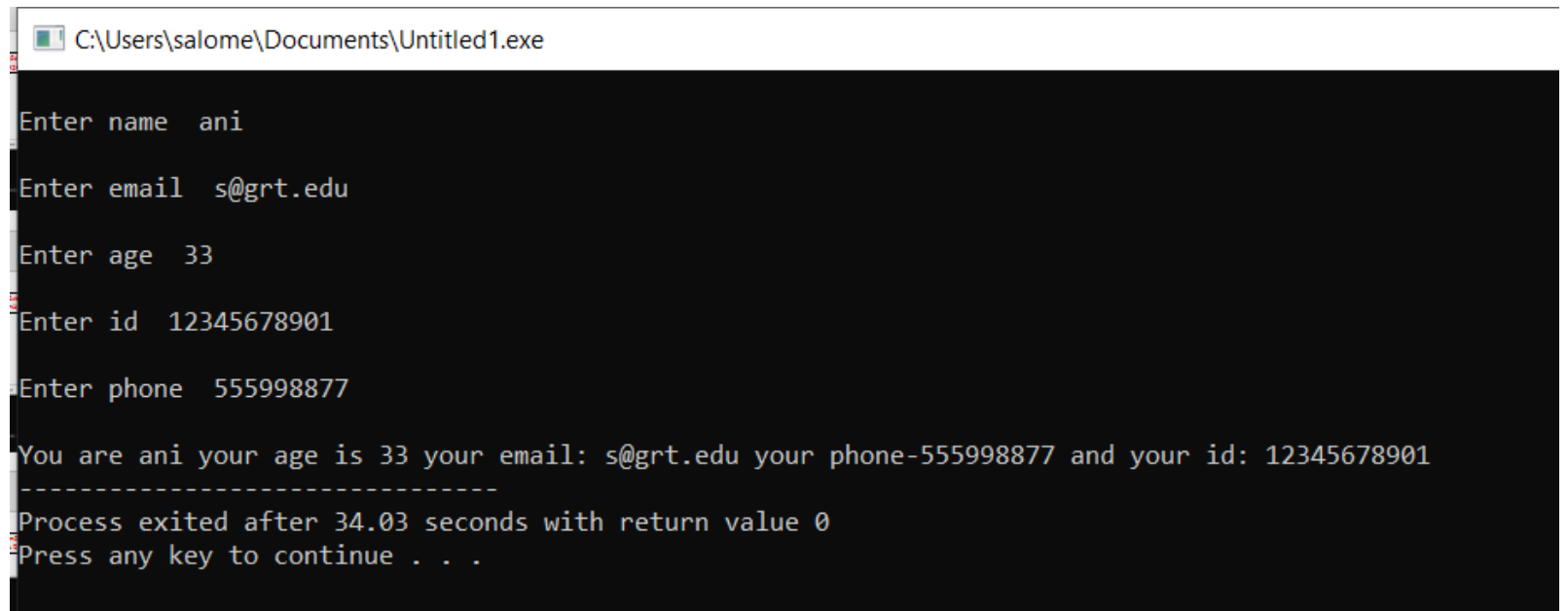


```
C:\Users\salone\Documents\Untitled1.exe
Enter name  salome
Enter email  s.e@gmail.com
Enter age  22
Enter id  19872637485
Enter phone  555888966

You are salome your age is 22 your email: s.e@gmail.com your phone-555888966 and your id: 19872637485
-----
Process exited after 25.67 seconds with return value 0
Press any key to continue . . .
```

```
#include <stdio.h>
struct Student{
    char name[50];
    int age;
    char email[30];
    unsigned long int phone;
    double id;
}student1, student2 ;
main (){
    printf("\nEnter name ");
    scanf("%s",&student1.name);
    printf("\nEnter email ");
    scanf("%s",&student1.email);
    printf("\nEnter age ");
    scanf("%d",&student1.age);
    printf("\nEnter id ");
    scanf("%lf",&student1.id);
    printf("\nEnter phone ");
    scanf("%d",&student1.phone);
    student2=student1;
    printf("\nYou are %s your age is %d your email: %s your phone-%d and your id: %0.f",student2.name, student2.age,
        student2.email, student2.phone, student2.id);
    return 0;
}
```

განსხვავებით მასივებისგან სტრუქტურებში
შესაძლებელია ერთი სტრუქტურის ცვლადი
მივანიჭოთ იგივე სტრუქტურის მეორე სხვა
ცვლადს



```
C:\Users\sалome\Documents\Untitled1.exe
Enter name  ani
Enter email  s@grt.edu
Enter age  33
Enter id  12345678901
Enter phone  555998877
You are ani your age is 33 your email: s@grt.edu your phone-555998877 and your id: 12345678901
-----
Process exited after 34.03 seconds with return value 0
Press any key to continue . . .
```

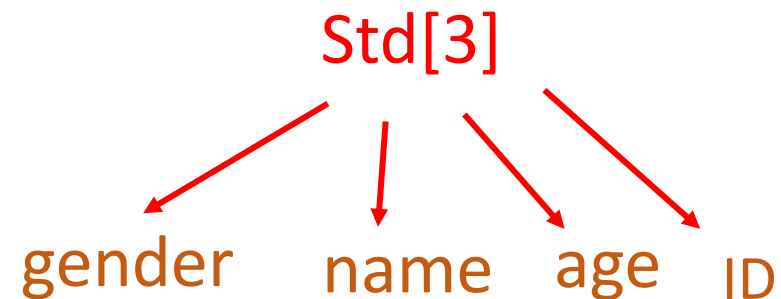
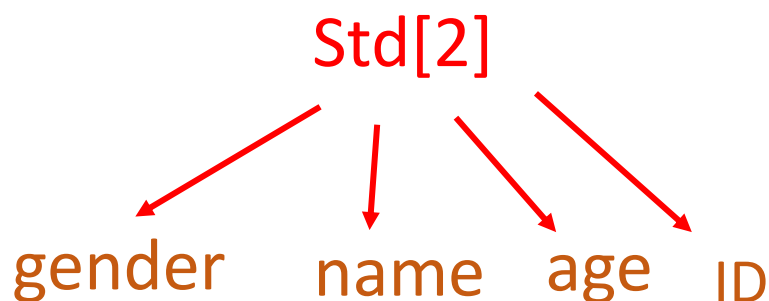
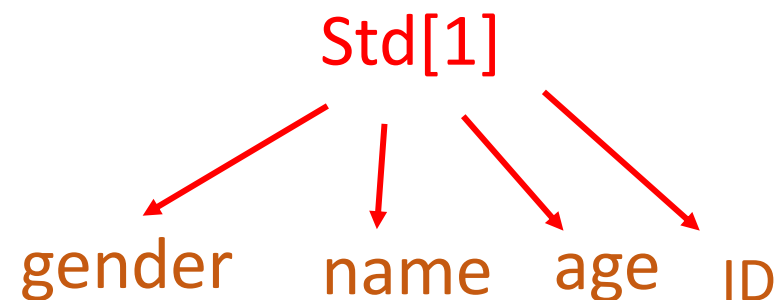
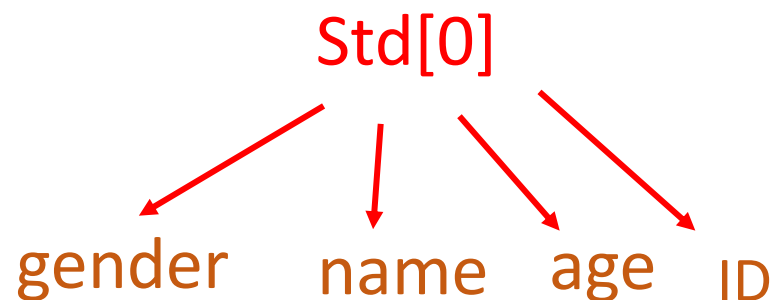



შექმენით პაციენტების სტრუქტურა, განსაზღვეთ
მასში შემავალი ველები, ველებში კონსოლიდან
შეინახეთ მონაცემები და შემდეგ დაბეჭდეთ ველებში
შენახული ინფორმაცია

სტრუქტურები

სტრუქტურის მასივის შექმნა

```
struct Student  
{  
    char name[25];  
    int age;  
    int ID;  
    char gender;  
} std[4];
```



სტრუქტურები

სტრუქტურის მასივთან მუშაობა

```
struct Student
{
    char name[25];
    int age;
    int ID;
    char gender;
};
```

```
struct Student std[4];
std[0].age = 19;
```



შექმენით ავტომანქანების რეგისტრაციის პროგრამა,
სადაც მომხმარებელი შეძლებს ავტომობილის
მონაცემების შეყვანას (მაგალითად ბრენდი, ფერი,
ძრავის სიმძლავრე, ავტომანქანის ნომერი).
სისტემაში შესაძლებელი იყოს 3 ავტომობილის
დარეგისტრირება. მონაცემების შეტანის
დასრულების შემდეგ პროგრამამ გამოიტანოს
დარეგისტრირებული მანქანების სია.

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

```
struct car{  
    char brand [20];  
    char color [10];  
    float power;  
    char number [10];  
  
    } cars[10];
```

```
int main(){  
    printf("Enter your favourite cars :) \n");  
  
    for (int i = 0; i < 3; i++){  
        printf("Car %d\n", i+1);  
        printf("Brand --> ");  
        scanf("%s", &cars[i].brand);  
        printf("Color --> ");  
        scanf("%s", &cars[i].color);  
        printf("Power --> ");  
        scanf("%f", &cars[i].power);  
        printf("Number --> ");  
        scanf("%s", &cars[i].number);  
    }  
    printf("\n\nYour favourite cars are \n");  
    printf("Car\t\t\tBrand\t\t\tColor\t\t\tPower\t\t\tNumber\n");  
    for (int i = 0; i < 3; i++){  
        printf("Car %d", i+1);  
        printf("\t\t\t%s", cars[i].brand);  
        printf("\t\t\t%s", cars[i].color);  
        printf("\t\t\t%.1f", cars[i].power);  
        printf("\t\t\t%s", cars[i].number);  
        printf("\n");  
    }  
  
    return 0;  
}
```



გაერთიანებები

გაერთიანებები (union) უზრუნველყოფენ განხვავებული მონაცემთა ტიპის ინფორმაციის შენახვას ერთიდაიგივე მეხსიერების მისამართზე. გაერთიანებების გამოიყენება მეხსიერების ეფექტურად სამართავად.

გაერთიანება განისაზღვრება შემდეგნაირად:

```
union „გაერთიანების სახელი“  
{  
    გაერთიანების წევრები;  
};
```

გაერთიანების განსაზღვრა

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

```
union Data {
    int i;
    float f;
    char str[20];
};
```

```
int main( ) {
```

```
    union Data data;
```

```
    printf( "Memory size occupied by data : %d\n", sizeof(data));
```

```
    return 0;
```

```
}
```

გაერთიანებამ აღნიშნული ველებით
მეხსიერებაში დაიკავა მაქსიმუმ 20 ბაიტი,
რადგან გაერთიანების წევრებს შორის ყველაზე
მეტი მეხსიერება სჭირდება str მასივს და მისი
მოცულობაა 20 ბაიტი

გაერთიანების ელემენტებზე წვდომა

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

```
union Data {
    int i;
    float f;
    char str[20];
};
```

```
int main( ) {
```

```
    union Data data;
```

```
    data.i = 10;
    data.f = 220.5;
    strcpy( data.str, "C Programming");
```

```
    printf( "data.i : %d\n", data.i);
    printf( "data.f : %f\n", data.f);
    printf( "data.str : %s\n", data.str);
```

```
    return 0;
```

```
}
```

მხოლოდ srt სტრუქტონში ჩაწერილი
მონაცემები დაიბეჭდება სწორად

გაერთიანების ელემენტებზე წვდომა

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

union Data {
    int i;
    float f;
    char str[20];
};

int main( ) {

    union Data data;

    data.i = 10;
    printf( "data.i : %d\n", data.i);

    data.f = 220.5;
    printf( "data.f : %f\n", data.f);

    strcpy( data.str, "C Programming");
    printf( "data.str : %s\n", data.str);

    return 0;
}
```

typedef

Typedef-ის გამოყენებით შესაძლებელია მონაცემთა ტიპს დაარქვით სასურველი სახელი

```
#include <stdio.h>

typedef unsigned char BYTE;

int main( ) {

    BYTE  b1 = 'c', b2 = 'p';

    printf("%c  %c", b1, b2);

    return 0;
}
```

typedef

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

```
typedef struct Books {
    char title[50];
    char author[50];
    char subject[100];
    int book_id;
} Book;
```

```
int main( ) {

    Book book;

    strcpy( book.title, "C Programming");
    strcpy( book.author, "Nuha Ali");
    strcpy( book.subject, "C Programming Tutorial");
    book.book_id = 6495407;

    printf( "Book title : %s\n", book.title);
    printf( "Book author : %s\n", book.author);
    printf( "Book subject : %s\n", book.subject);
    printf( "Book book_id : %d\n", book.book_id);

    return 0;
}
```

Typedef ასევე გამოყენება სტრუქტურებთან



შექმენით სტუდენტების რეგისტრაციის პროგრამა, სადაც მომხმარებელი შეძლებს სტუდენტების მონაცემების შეყვანას (მაგალითად სახელი-გვარი, ასაკი, სქესი და ა.შ). სისტემაში შესაძლებელი იყოს იმდენი სტუდენტის რეგისტრაცია რამდენიც მომხმარებელს სურს. მონაცემების შეტანის დასრულების შემდეგ პროგრამამ გამოიტანოს დარეგისტრირებული სტუდენტების სია.



შექმენით სტუდენტების რეგისტრაციის პროგრამა, სადაც მომხმარებელი შეძლებს სტუდენტების მონაცემების შეყვანას (მაგალითად სახელი-გვარი, ასაკი, სქესი და ა.შ). სისტემაში შესაძლებელი იყოს იმდენი სტუდენტის რეგისტრაცია რამდენიც მომხმარებელს სურს. მონაცემების შეტანის დასრულების შემდეგ პროგრამამ გამოიტანოს დარეგისტრირებული სტუდენტების სია.



შექმენით ფეხბურთის გუნდების რეგისტრაციის
პროგრამა, სადაც მომხმარებელი შეძლებს
ფეხბურთის გუნდების მონაცემების შეყვანას
(მაგალითად ქვეყანა, გუნდის დასახელება, კაპიტანი
და ა.შ). სისტემაში შესაძლებელი იყოს იმდენი
გუნდის რეგისტრაცია რამდენიც მომხმარებელს
სურს. მონაცემების შეტანის დასრულების შემდეგ
პროგრამამ გამოიტანოს დარეგისტრირებული
გუნდების სია.



შექმენით მობილური ტელეფონების რეგისტრაციის
პროგრამა, სადაც მომხმარებელი შეძლებს იმ
მობილური ტელეფონების მონაცემების შეყვანას
(მაგალითად ბრანდი, სერიული ნომერი, ტევადობა
და ა.შ), რომლებიც მას მოსწონს. სისტემაში
შესაძლებელი იყოს იმდენი ტელეფონის
რეგისტრაცია რამდენიც მომხმარებელს სურს.
მონაცემების შეტანის დასრულების შემდეგ
პროგრამამ გამოიტანოს დარეგისტრირებული
ტელეფონების სია.



დავწეროთ პროგრამა, რომელიც დაითვლის
მომხმარებლის მიერ შემოტანილ წინადადებაში
სიტყვების რაოდენობას



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

int main( ) {

    char word[200];
    printf("type the message ");
    gets(word);

    int counter = 0;

    for(int i=0; i<strlen(word); i++){
        if(word[i] == ' ')
            counter++;
    }

    printf("your message has %d words", counter+);

    return 0;
}
```



დავწეროთ პროგრამა, რომელიც მომხმარებლის მიერ
შემოტულ წინადადებაში ყველა სიტყვას დაიწყებს
დიდი ასოთი

გმადლობთ ყურადღებისთვის!