



გამოცდის ფორმატი

\*მონიშნეთ გამოცდის ფორმატი (მიუთითეთ ✓)

დახურული წიგნი	
ღია წიგნი	✓

\*ღია წიგნის შემთხვევაში მონიშნეთ გამოცდაზე ნებადართული ელემენტები (მიუთითეთ ✓)

საღიგებულო მასალები (პრეზენტაცია და სხვა)	
ელექტრონული წიგნები	✓
წიგნები	
კონსპექტები	
ლექსიკონი	
კალკულატორი	
ლექტორი/პლანშეტი	

\* გამოცდის ჩატარების წესი იხილეთ „დესკტოპზე“ საქალაქო Exam materials

საგამოცდო საკითხების ფორმა  
ვარიანტი # 1

სკოლა/საგანმანათ ლებლო პროგრამა	მათემატიკა და კომპიუტერული მეცნიერება	სტუდენტის მიერ მიღებული ქულა	
საგანი	პროგრამირების პარადიგმები		
ლექტორი	შ. ლვინევაძე		
კურსი	II		
ჯგუფი			
გამოცდის ფორმა	ღია წიგნი		
გამოცდის ხანგრძლივობა	2 საათი		
მაქსიმალური ქულა	120		
სტუდენტის სახელი და გვარი:			

შუალედური გამოცდა 2  
პარადიგმებში  
2016, 5 დეკემბერი

1 30 ქულა	2 30 ქულა	3 60 ქულა

ამოცანა 1. ასემბლერი (30 ქულა)



შექმენით assembly.txt ფაილი და დაწერეთ ასემბლერის კოდი fairytale ფუნქციისთვის. ცალკე გამოყავით თითოეული ხაზის შესაბამისი კოდი.

```
typedef struct Sizmara{
    int sun;
    struct Sizmara *moon;
    char stars[8];
}Sizmara;

typedef struct Komble{
    int kombal;
    struct Komble **next;
    struct Sizmara niko;
}Komble;

Komble *fairytale(Sizmara *si, Komble ko){
    si->moon->sun =
        *(int*)&((Sizmara *) (ko.next + ko.kombal))->stars[4]);
    ko.next = malloc(2);
    return *ko.next;
}
```

ამოცანა 2. ონავარი სტუდენტები (30 ქულა)



მოცემული გაქვთ სტუდენტის სტრუქტურა და სტრუქტურის მასივი. თქვენი ამოცანაა 3 მარტივი ფუნქციის დაწერა

`sort_by_name()` ფუნქცია ასორტირებს სტუდენტებს სახელების ზრდის მიხედვით.  
`sort_by_age()` ფუნქცია ასორტირებს სტუდენტებს ასაკის ზრდის მიხედვით. ორივე ფუნქციაში დასორტირებისთვის უნდა გამოიყენოთ `qsort` ფუნქცია სტანდარტული ბიბლიოთეკიდან.

`isolder()` ფუნქციას გადაეცემა ასაკი და უნდა დაბეჭდოს ამ ასაკზე უფროსი სტუდენტები.

`main` ფუნქცია და სხვა დამხმარე ფუნქციები უკვე წერია `student.c` ფაილში. ცვლილებები შეგიძლიათ შეიტანოთ მხოლოდ და მხოლოდ მითითებულ ფუნქციებში (`sort_by_name`, `sort_by_age`, `isolder`), ასევე შეგიძლიათ შექმნათ დამხმარე ფუნქციები.

ეს პროგრამა გასწორდება ტესტებზე ამიტომ გთხოვთ ჩააბაროთ კომპილირებადი კოდი.

შეგახსენებთ `qsort`-ის ჰედერს.

```
void qsort(void *base, size_t nel, size_t width, int (*compar)(const void *, const void *));
```

### ამოცანა 3. Circle სტრუქტურა (60 ქულა)



Circle სტრუქტურის საშუალებით შესაძლებელია int ტიპის ელემენტების სიის შენახვა და შემდეგ მნიშვნელობების ამოღება. სტრუქტურა ჰგავს Vector-ს, თუმცა რეალიზაცია საკმაოდ განსხვავებულია, Circle სტრუქტურა ელემენტების შესანახად იყენებს ციკლურ ბმულ სიას.

სტრუქტურას გააჩნია შემდეგი ფუნქციები

`void CircleInsert(Circle * c, int index, int value)` - საშუალებას გვაძლევს ელემენტი, მნიშვნელობით `value`, ჩავამატოთ `index`-ში ადგილას, სადაც `n` სტრუქტურაში არსებული ელემენტების რაოდენობაა. ყურადღება მიაქციეთ იმ ფაქტს რომ ბმული სია ციკლურია და `n` ინდექსზე მყოფი ელემენტი იგივე ნულოვანი ელემენტია, `n+1` ინდექსზე მყოფი პირველი და ა.შ.

ელემენტის დამატებისას ხდება მისი და მის ირგვლივ მყოფი ელემენტების გასაშუალოება გასაშუალოება. მაგალითად თუკი დავამატეთ ელემენტი ინდექსზე 2, მაშინ 1, 2 და 3 ინდექსზე მყოფი ელემენტების მნიშვნელობები შეიცვლებიან და მათ მაგივრად დაიწერება მათი საშუალო არითმეტიკული.

`int CircleGetNth(Circle * c, int index)` - საშუალებას გვაძლევს დავაბრუნოთ `index` ინდექსზე მყოფი ელემენტი.

`int CircleSize(Circle * c)` - საშუალებას გვაძლევს დავაბრუნოთ სტრუქტურაში არსებული ელემენტების რაოდენობა.

ფაილებში `circle.c` და `circle.h` მოცემული გაქვთ სტრუქტურის რეალიზაცია. ფუნქციონალი გამართულად მუშაობს თუ მეთოდების გამოძახება მოხდება მხოლოდ ერთი ნაკადიდან.

თქვენი ამოცანა გადააკეთოთ კოდი ისე, რომ სტრუქტურის ფუნქციონალი გამართულად მუშაობდეს პარალელურად რამდენიმე ნაკადიდან გამოძახების შემთხვევაშიც.

რეალიზაცია უნდა იყოს მაქსიმალურად ოპტიმიზირებული, `CircleInsert` ფუნქციის პარალელური შესრულებები არ უნდა აბრკოლებდნენ ერთმანეთს თუკი ჩამატება ხდება ორით მაინც დამორეზულ ადგილას.

`CircleGet` ფუნქციის პარალელური შესრულებები არ უნდა აბრკოლებდნენ ერთმანეთს არც ერთ შემთხვევაში.

შენიშვნა: ამოცანა არ გასწორდება ტესტებზე, ამიტომ კომპილირებადი კოდის დაწერა არ არის აუცილებელი.



ფუნქციების ჰედერები, რომლებიც შეიძლება დაგჭირდეთ ამოცანების გადაჭრისას:

```
void *memcpy(void *dest, const void *src, size_t n);  
void *memmove(void *dest, const void *src, size_t n);  
void *malloc(size_t size); void *realloc(void *ptr, size_t size);  
void free(void *ptr);
```

```
size_t strlen(const char *s);  
char *strcpy(char *dest, const char *src);  
char *strncpy(char *dest, const char *src, size_t n);  
char *strdup(const char *s);  
char *strcat(char *dest, const char *src);
```

command prompt-ის გამოსაყენებლად

1. დააჭირეთ windows ღილაკს ეკრანის მარცხენა ქვედა კუთხეში
2. ძეგნის ფანჯარაში აკრიბეთ command prompt
3. დააკლიკეთ მაუსი command prompt-ის იკონს.
4. ფოლდერში ინფორმაციის ნახვისთვის გამოიყენეთ ბრძანება DIR(იგივე ls)
5. ფოლდერის შეცვლისთვის გამოიყენეთ cd

მუშაობის დასრულების შემდეგ დესკტოპზე შექმენით ფოლდერი თქვენი მეილის პრეფიქსის სახელით(მაგ gboch15) და ამ ფოლდერში ჩადეთ თქვენი ფაილები.  
ფოლდერში უნდა იყოს მხოლოდ შემდეგი ფაილები:

1. student.c
2. circle.h
3. circle.c
4. assembly.txt