



გამოცდის ფორმატი

*მონიშნეთ გამოცდის ფორმატი (მიუთითეთ ✓)

დახურული წიგნი	
ღია წიგნი	✓

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

სალექციო მასალები (პრეზენტაცია და სხვა)	
ელექტრონული წიგნები	✓
წიგნები	
კონსპექტები	
ლექსიკონი	
კალკულატორი	
ლექტოპი/პლანშეტი	

* გამოცდის ჩატარების წესი იხილეთ „დესკტოპზე“ საქალაქო Exam materials

საგამოცდო საკითხების ფორმა ვარიანტი # 1

სკოლა/ საგანმანათლებლო ლო პროგრამა	მათემატიკა და კომპიუტერული მეცნიერება	სტუდენტის მიერ მიღებული ქულა	
საგანი	პროგრამირების პარადიგმები		
ლექტორი	შ. ღვინეფაძე		
კურსი	II		
ჯგუფი			
გამოცდის ფორმა	ღია წიგნი		
გამოცდის ხანგრძლივობა	3 საათი		
მაქსიმალური ქულა	120		
სტუდენტის სახელი და გვარი:			

სახელი:

ქულა:



შუალედური გამოცდა
პარადიგმებში
2017, 24 ოქტომბერი 14:40 – 16:40

1 50 ქულა	2 70 ქულა	სულ

შეასრულეთ შემდეგი ინსტრუქციები, წინააღმდეგ შემთხვევაში შესაძლოა თქვენი ნაშრომი არ შეფასდეს.

1. ჩამოტვირთეთ problems ფოლდერი თქვენს დესკტოპზე. მასში უნდა იყოს 2 ფოლდერი problem1 და problem2. თითოეულში კი შესაბამისი ფაილები.
2. ცვლილებები შეიტანეთ დავალების პირობით მითითებულ ფაილებში.
3. ის ფაილები, რომელშიც ცვლილებები შეიტანეთ დააარქივეთ, არქივს სახელად დაარქვით თქვენი მეილის პრეფიქსი, მაგალითად gboch12.rar. არქივში უნდა იყოს მხოლოდ 2 ფაილი
1. find_score.c
2. spell_correct.c
4. ვებ ბრაუზერში გახსენით მისამართი <http://192.168.210.5> და ატვირთეთ არქივი.

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

ამოცანა 1. მინიმუმი(30 ქულა)

მოცემული გაქვთ გენერიკ მასივი, თქვენი მიზანია ამ მასივში იპოვოთ მინიმალური ელემენტი. ელემენტების შედარებისთვის გადმოგეცემათ შედარების



ფუნქცია, რომელიც აბრუნებს 0-ს თუკი ორი ელემენტი ტოლია -1 თუკი პირველი ელემენტი პატარაა მეორეზე და 1-ს წინააღმდეგ შემთხვევაში.

base - გენერიკ მასივის მისამართი

n - მასივში ელემენტების რაოდენობა

size - ელემენტის ზომა ბაიტებში

cmp - შედარების ფუნქცია.

თქვენმა ფუნქციამ უნდა დააბრუნოს მინიმალურ ელემენტზე მიმთითებელი.

void* minimum(void *base, int n, int size, int(*cmp)(const void *, const void *))

კომპილაციისთვის შეგიძლიათ გამოიყენოთ ბრძანება

> gcc minimum.c minimum_test.c

მიიღებთ a.exe ფაილს, რომელიც შეგიძლიათ გაუშვათ

> a.exe

ამოცანა 2. სორტირება(30 ქულა)

მოცემული გაქვთ გენერიკ მასივი, თქვენი მიზანია ამ მასივის ზრდადობით დასორტირება. ჩათვალეთ, რომ ფუნქცია minimum (ამოცანა 1-დან)

იმპლემენტირებულია და გამოიყენეთ selection sort ალგორითმი: minimum ფუნქციით აირჩიეთ მინიმალური ელემენტი და პირველ ელემენტთან გაუცვალეთ ადგილი. შემდეგ იტერაციაზე აირჩიეთ მინიმუმი მეორე ელემენტიდან (minimum ფუნქციის დახმარებით), ადგილი გაუცვალეთ მეორე ელემენტთან და ა.შ.

my_sort ფუნქცია არაფერს არ აბრუნებს, ის გადაცემულ მასივს ასორტირებს.

base - გენერიკ მასივის მისამართი

n - მასივში ელემენტების რაოდენობა

size - ელემენტის ზომა ბაიტებში

cmp - შედარების ფუნქცია.

void my_sort(void *base, int n, int size, int(*cmp)(const void *, const void *))

კომპილაციისთვის შეგიძლიათ გამოიყენოთ ბრძანება

> gcc done_minimum.o my_sort.c my_sort_test.c

მიიღებთ a.exe ფაილს, რომელიც შეგიძლიათ გაუშვათ

> a.exe

ამოცანა 3. სტრინგების სორტირება(30 ქულა)

თქვენი მიზანია my_sort ფუნქციის გამოყენებით დაასორტიროთ სტრინგები.

ამისათვის დანერეთ sort_strings ფუნქცია, რომელიც არაფერს არ აბრუნებს,

არამედ გადაცემულ მასივს ასორტირებს. ჩათვალეთ, რომ my_sort ფუნქცია უკვე იმპლემენტირებულია.

strings - სტრინგების მასივი

n - მასივში ელემენტების რაოდენობა

void sort_strings(char** strings, int n)



კომპილაციისთვის შეგიძლიათ გამოიყენოთ ბრძანება

```
> gcc done_my_sort.o sort_strings.c sort_strings_test.c
```

მიიღებთ a.exe ფაილს, რომელიც შეგიძლიათ გაუშვათ

```
> a.exe
```

ფუნქციების პედერები, რომლებიც შეიძლება დაგჭირდეთ ამოცანების გადაჭრისას:

```
void *memcpy(void *dest, const void *src, size_t n);
```

```
int memcmp(const void *s1, const void *s2, 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);
```

```
int strcmp(const char *s1, const char *s2);
```

```
char *strncpy(char *dest, const char *src, size_t n);
```

```
char *strdup(const char *s);
```

```
char *strcat(char *dest, const char *src);
```