



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	
ლექტორი/პლანშეტი	

* გამოცდის ჩატარების წესი იხილეთ „დესკტოპზე“ საქალაქოში Exam materials

საგამოცდო საკითხების ფორმა ვარიანტი # 1

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

შუალედური გამოცდა
პარადიგმებში
2016, 28 ოქტომბერი 17:00 – 19:00

1 50 ქულა	2 70 ქულა	სულ

ამოცანა 1. float-ის წარმოდგენა(50 ქულა)

თქვენი ამოცანაა float ტიპის ცვლადის ტექსტურ ფორმატში წარმოდგენა.

შეგახსენებთ რომ float არის ტიპი რომელიც მეხსიერებაში შემდეგი სახით ინახება:



float ტიპის ცვლადი იკავებს 32 ბიტის.
პირველი ბიტი არის ნიშნის ბიტი - sign
შემდეგი 8 ბიტი არის ექსპონენტის ბიტები - exp
დანარჩენი 23 ბიტი კი 1.xxx რიცხვს აღწერს -

უნდა დაწეროთ ფუნქცია float_to_text რომელსაც გადაეცემა float ცვლადი და რომელიც აბრუნებს C-სტრინგს - ამ ცვლადის ტექსტურ ვარიანტს.

დაბრუნებული ტექსტის ფორმატი უნდა იყოს შემდეგი:

[sign][1.xxx] * 2^[127-exp]

სადაც [sign]-ის მაგივრად იწერება სიმბოლო '-' თუკი რიცხვი უარყოფითია, თუკი რიცხვი დადებითია მაშინ არაფერი არ იწერება.

[1.xxx]-ის მაგივრად იწერება 1. - სულ 25 სიმბოლო. მიაქციეთ ყურადღება რომ სიმბოლოები არის '0' ან '1'.

შემდეგ მოდის სფეისი, სიმბოლო '*', სფეისი, სიმბოლო '2', სიმბოლო '^'

ხოლო [127-exp]-ის მაგივრად იწერება პირდაპირ მნიშვნელობა 127-exp, რომელიც შეიძლება იყოს როგორც დადებითი ისე უარყოფითი.

მაგალითად:

float_to_text(1.5) უნდა აბრუნებდეს C-სტრინგს "1.100000000000000000000000 * 2⁰"

float_to_text(0.15625) უნდა აბრუნებდეს C-სტრინგს "1.010000000000000000000000 * 2⁻³"

float_to_text(-7.333) უნდა აბრუნებდეს C-სტრინგს "-1.11010101010011111110000 * 2²"

მოცემული გაქვთ FloatToTextTest.c ფაილი. ეს ფაილი გადმოაკოპირეთ თქვენს კომპიუტერში და შეიტანეთ მასში ცვლილებები. ამ ფაილში შეცვალეთ მხოლოდ ფუნქცია float_to_text. ცხადია დეკომპოზიცია შეგიძლიათ გააკეთოთ. არ გამოიყენოთ გლობალური ცვლადები. პროგრამა უნდა დაიწეროს C-ში.

კომპილაციისთვის შეგიძლიათ გამოიყენოთ ბრძანება

> gcc FloatToTextTest.c

მიიღებთ a.exe ფაილს, რომელიც შეგიძლიათ გაუშვათ

> a.exe

ამოცანა 2. Generic Generic List(70 ქულა)

თქვენი ამოცანაა ყველაზე ჯენერიკ ლისტის დაწერა. სტრუქტურა ლისტი ძალიან ჰგავს ვექტორს იმ განსხვავებით რომ ის არაერთგვაროვანია ანუ მასში შეიძლება



სხვადასხვა ტიპის ელემენტების დამატება ერთდროულად. მაგალითად int-ების, float-ების, char*-ების და ა.შ.

უნდა მოიფიქროთ List სტრუქტურის შინაარსი და უნდა გააკეთოთ შემდეგი ფუნქციების იმპლემენტაცია:

```
void ListNew(List *l);
```

ფუნქცია ინიციალიზებას უკეთებს ლისტ სტრუქტურას. უნდა სრულდებოდეს $O(1)$ დროში.

```
void ListAppend(List *l, void *elemAdr, int elemSize);
```

ფუნქციას გადაეცემა ელემენტზე მიმთითებელი და ელემენტის ზომა და ის ლისტის ბოლოში ამატებს ამ ელემენტს. გაითვალისწინეთ, რომ ფუნქციამ უნდა გადააკოპიროს გადმოცემული ელემენტი. შესაბამისად, თუკი მომხმარებელი elemAdr მისამართზე მყოფ ელემენტს შეცვლის append-ის შემდეგ, თქვენს სტრუქტურაში მყოფი ელემენტი არ უნდა შეიცვალოს. ფუნქცია უნდა სრულდებოდეს $O(1)$ დროში.

```
void ListRemove(List *l, int index);
```

ფუნქცია შლის index ადგილას მყოფ ელემენტს. შესაბამისად ლისტში ელემენტების რაოდენობა იკლებს ერთით. ფუნქცია უნდა სრულდებოდეს $O(n)$ დროში

```
void ListGet(List *l, int index, void *elemAdr);
```

ფუნქციამ elemAdr მისამართზე უნდა ჩაწეროს index ინდექსზე მყოფი ელემენტის მნიშვნელობა. ფუნქცია უნდა სრულდებოდეს $O(1)$ დროში.

```
void ListDispose(List *l);
```

ფუნქცია ასუფთავებს თქვენს მიერ გამოყოფილ მეხსიერებას. ფუნქცია უნდა სრულდებოდეს $O(n)$ დროში.

(ბონუსი 15 ქულა)

```
void ListInsert(List *l, int index, void *elemAdr);
```

ფუნქციამ index ინდექსზე ამატებს ახალ ელემენტს, რომელსაც მიუთითებს elemAdr მიმთითებელი. index ინდექსზე მაღლა მდგომი ელემენტების ინდექსები ერთით უნდა გაიზარდოს. ლისტში ელემენტების რაოდენობაც იზრდება ერთით.

(ბონუსი 10 ქულა)

```
void ListPrint(List *l, void(*PrintFn)(void*));
```



ფუნქციას გადაეცემა PrintFn ფუნქცია რომელსაც შეუძლია ელემენტის დაბეჭდვა კონსოლში. იმისათვის რომ PrintFn-მა სწორად დაბეჭდოს ელემენტი საჭიროა მას გადასცეთ ამ ელემენტზე მიმთითებელი(ზუსტად იგივე მისამართი რაც გადმოეცემა ListAppend ან ListInsert ფუნქციებს). ListPrint-მა უნდა დაბეჭდოს ლისტის ყველა ელემენტი კონსოლში.

გაითვალისწინეთ, რომ არ არის აუცილებელი მომხმარებლის მიერ შექმნილი ელემენტების მეხსიერებიდან წაშლა. გაასუფთავეთ მხოლოდ თქვენს მიერ შექმნილი ცვლადები.

მოცემული გაქვთ list.h, list.c და list_test.c ფაილები. გადმოაკოპირეთ ისინი თქვენს კომპიუტერში. ცვლილებები შეგიძლიათ შეიტანოთ მხოლოდ list.h და list.c ფაილებში.

კომპილაციისთვის გამოიყენეთ ბრძანება

```
> gcc list.c list_test.c
```

მიიღებთ a.exe ფაილს, რომელიც შეგიძლიათ გაუშვათ

```
> a.exe
```

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. ძებნის 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. FloatToTextTest.c
2. list.h
3. list.c
4. list_test.c