

# დავალება 1: c++

პირველი დავალების მიზანია შეეჩვიოთ პროგრამირების ერთ ერთ ენას C++ს. გაერკვიოთ თუ როგორ მუშაობს ფუნქციები, პრიმიტიული ტიპები, სტრინგები და გაეცნოთ თქვენთვის ახალ თემას რეკურსიას.

## ნაწილი პირველი. cs106b

თქვენი პირველი ამოცანაა დააყენოთ C++ ის კომპილატორი. მათი დაყენების ინსტრუქცია წერია შემდეგ ჰენდაუტებში იმისდა მიხედვით თ რომელ ოპერაციულ სისტემას იყენებთ.

- **Handout #04M. Downloading Xcode on the Macintosh**
- **Handout #04P. Downloading Visual Studio for Windows**
- **Handout #04L. Getting Started on Linux**

მას შემდეგ რაც გემოთ ხსენებულიდან ერთ ერთს დააყენებთ შედით საიტზე და გადმოწერეთ **starter** ფაილი იმისდა მიხედვით თუ რომელ სისტემას იყენებთ. დავალება 1 შეიცავს 5 სხვადასხვა პროექტს. გახსენით პროექტი სახელად **0-warmup**. თქვენი ამოცანაა უბრალოდ გაუშვათ ეს პროექტი. ამ პროექტში სპეციალურად 2 შეცდომაა დაშვებული, თქვენი მიზანია გამოასწოროთ ეს შეცდომები და გაუშვათ პროგრამა. ამ ამოცანის მიზანია გაეკვიოთ თუ როგორ გვეუბნება კომპილატორი დაშვებულ შეცდომებს პროექტში და როგორ უნდა ვიპოვოთ და გამოვასწოროთ.

როდესაც გამოასწოვებთ კომპილაციის შეცდომებს გაუშვით პროექტი. როდესაც პროგრამა გაეშვება მოგთხოვთ თქვენი სახელის შეყვანას. შეიყვანეთ თქვენი სახელი და ის დაგიბეჭდავთ თქვენი სახელის ჰეშს ანუ რიცხვს რომელიც თქვენ სახელს შეესაბამება. მოგვიანებით ჩვენ ვისაუბრებთ ჰეშებზე და რისთვის გამოიყენება მაგრამ ამჯერად მხოლოდ გაუშვით პროგრამა და ნახეთ ჰეშ კოდი.

## ნაწილი მეორე.

სურათიდან ჩანს რომ როდესაც სამკუთხედში გემოდან ქვემოთ ვმოძრაობთ იზრდება  $n$  ხოლო როცა მარცხნიდან მარჯვნივ იზრდება  $k$ . მოდით ეხლა გვდა სურათში  $C$  ების

მაგივრად ჩავსვით მათი რიცხვითი მნიშვნელობები და მივიღებთ შემდეგ სურათს.

		1						
	1		1					
	1	2		1				
	1	3	3		1			
	1	4	6	4		1		
	1	5	10	10	5	1		
	1	6	15	20	15	6	1	
	1	7	21	35	35	21	7	1

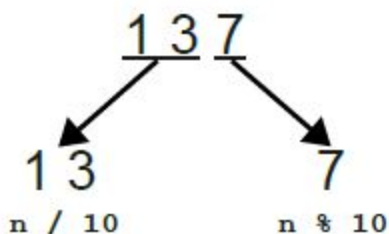
პასკალის სამკუთხედს აქვს ერთი საინტერესო თვისება რომ ნებისმიერ წვეროში მნიშვნელობა გოლია მის ზემოთ მდებარე 2 წვეროს მნიშვნელობების ჯამის. მაგალითად ამ შემთხვევაში, სურათზე მწვანედ გაფერადებულ წვეროს მნიშვნელობა 15 (რომელიცაა  $C(6,2)$ ) გოლია მის ზემოთ მყოფი ორი წვეროს 5 ის და 10 ის ჯამი. ამ ფაქტზე დაყრდნობით თქვენი ამოცანაა  $C(n,k)$  ს დათვლა რეკურსლიულად.

დეალეზა 3: რიცხვითი გარდაქმნის იმპლემენტაცია.

strlib.h ინტერფეისის მოყვება ორი მეთოდი რომელიც ახდენს ინტის სკრინგში დ სკრინგის ინტში გადაყვანას.

```
string integerToString(int n);
int stringToInteger(string str);
```

მაგალითად integerToString(1927) დაგვიბრუნებს სკრინგს "1927" და stringToInteger("-42") დაგვიბრუნებს ინტს -42ს. ამ დავალების მიზანია თქვენ დაწეროთ თქვენი ორი რეკურსული მეთოდი intToString და stringToInt რომელიც გააკეთებს იგივეს. რეკურსული მეთოდისთვის უნდა გამოვიყენოთ შემდეგი თვისება, ჩვენ შეგვიძლია მათემატიკური ოპერაციებით ნაშთით და გაყოფით გავყოთ ინტი ორ ნაწილად ისე როგორც ეს სურათზეა ნაჩვენები



სხვა სიგეყებით რომ ვთქვათ შეგვიძლია ინტიდან გამოვყოთ ბოლო ციფრი. თუ გამოვიდახებთ რეკურსიას დარჩენილ ნაწილზე ამ შემთხვევაში 13 ზე და გამოყოფილ ციფრს 7ს მივაწებებთ `character` ის სახით მივიღებთ ამოხსნას.

ასევე შეგვეძლო გამოგვეყო პირველი ციფრი და არა ბოლო მაგრამ ამ შემთხვევაში კოდის იმპლემენტაცია უფრო რთული იქნებოდა.

თქვენი ამოხსნა მარტო რეკურსიული უნდა იყოს და რეკურსიაში არ უნდა იყენებდეს `for` და `while` ციკლებს. ასევე არ შეგიძლიათ სხვა ბიბლიოთეკების კონვერტაციების გამოყენება მაგალითად ამ შემთხვევაში არ შეგიძლიათ `strlib.h` ის ზემოთ ხსენებული მეთოდების გამოყენება რადგან დავალებას ამრი დაეკარგება.

ამ ამოცანის კეთებისას უნდა გაითვალისწინოთ რამდენიმე რაღაც: 10 ზე დანაშთვით მიღებული ბოლო ციფრი არის ინტი და არა სტრინგი. შესაბამისად მოგიწევთ ამ ინტის `character` ად გადაგქცევა რომელშიც შეგიძლიათ გამოიყენოთ კასტი მაგ: `(char(ch + '0'))` . თუ თქვენ გინდათ ამ `character` ის სტრინგად გადაგქცევა უნდა გამოიყენოთ კონკატერნაცია შემდეგი სახით `string() + ch`. დაუკვირდით რომ ჯავაში როდესც გვინდოდა სტრინგად გადაგქცევა ვიყენებდით ესეთ ფანდს `"" + ch` მაგრამ `C++` ის კომპილატორი ამას ვერ აღიქვამს და შეიძლება უცნაური რაღაცეები გააკეთოდს მაგალითად დააბრუნოს ნაგავი ან დაქრეშოს პროგრამა.

ასევე გაითვალისწინეთ კონკრეტული შემთხვევა `intToString(0)` რომელმაც უნდა დააბრუნოს სტრინგის სახით `"0"` და არა ცარიელი სტრინგი, ამისთვის წესით სპეციალური შემთხვევის დაწერა მოგიწევთ.

ასევე გაითვალისწინეთ რომ `stringToInt` მეთოდი უარყოფით რიცხვებზეც უნდა მუშაობდეს.

შეგიძლიათ სხვადასხვანირად დახლიჩოთ ინტი და არა ისე როგორც ზემოთ როცა სულ ბოლო ციფრზე ვხლიჩავთ, მაგრამ გაითვალისწინეთ რომ ესე დაწერა არის ყველაზე მარტივი.

ჩვენი რჩევა იქნება რომ პროგრამის დაწერის შემდეგ გაგესტოთ ეს პროგრამა ბევრ სხვადასხვა შემთხვევაზე რამეთუ აღვილია რომელიმეს გამორჩენა.

#### **პრობლემა 4: The Flesch-Kincaid Grade-Level Test**

ბევრი პროგრამა მათ შორის `Microsoft Word` იც აგებს ევრისტიკას და აფასებს თუ რამდენად რთულია მოცემული ტექსტი. ეს ეხმარება ავტორებს რო სწორად შეარჩიონ ტექსტი თავიანთი აუდიტორიისთვის. ასევე ძალიან სასარგებლოა მასწავლებლებისთვისაც.

ერთ ერთი ტექსტის შეფასების ტესტი არის Flesch-Kincaid grade level test რომელიც ტექსტს უწერს ქულას. ეს ქულა განსაზღვრავს სირთულეს და იმას თუ რა დონეზე უნდა იყოს ადამიანი რომ გაიგოს ეს ტექსტი. უფრო კონკრეტულად კი ეს ტესტი მუშაობს შემდეგნაირად. მოცემულ ტექსტში ითვლის რამდენი სიტყვაა და რამდენი წინადადება, ასევე ითვლის მარცვლების რაოდენობას. შემდეგ კი იყენებს ფორმულას რომელიც სურათზეა

$$Grade = C_0 + C_1 \left( \frac{\text{num words}}{\text{num sentences}} \right) + C_2 \left( \frac{\text{num syllables}}{\text{num words}} \right)$$

მოცემული .

სადაც C0 C1 და C2 კონსტანტებია და აქვს შემდეგი რიცხვითი მნიშვნელობები.

C0 = -15.59 C1 = 0.39 C2 = 11.8

Grade ის რიცხვითი მნიშვნელობა კი განსაზღვრავს სირთულეს მაგალითად : თუ grade გოლია 5.15 ეს ნიშნავს რომ ტექსტი გათვლილია მე5 კლასის მოსწავლისთვის ხოლო თუ 15 ს უდრის ნიშნავს რომ სკულენცებისთვისა.

თქვენი მიზანია დაწეროთ პროგრამა რომელიც გამოითვლის ამ ალგორითმით grade\_ს. ტექსტი მოცემულია ფაილის სახით შესაბამისად პროგრამამ გაშვებისთანავე მომხმარებელს უნდა მოთხოვოს ფაილის სახელის შეყვანა მანამა სანამ არ შეიყვანს მომხმარებელი სწორად ფაილის სახელს.

როცა გავხსნით ფაილს შემდეგი ამოცანაა დავეოთ სიტყვებად და პუნქტუაციის ნიშნებად ამისთვის დაგვეხმარება TokenScanner კლასი. ჩვენ TokenScanner ის ყველა ფუნქციონალს ლექსიაზე არ ავხსნით ამიტომ თქვენ თვითონ მოგიწევთ ამ კლასში გარკვევა.

1. უფრო კონკრეტულად უნდა გაერკვიოთ თუ როგორ უნდა გადაახტეთ პირდაპირ whitespace ებს რომ შემდეგ არ მოგიწიოთ ამის ჰენდლინგი.
2. წაიკითხოთ პირდაპირ ფაილიდან რომელიც რომელიც საგრძნობლად გაამარტივებს ლოგიკას.
3. გოკენაიზერით სტრინგი "isn't" უნდა ამოიღოთ როგორც "isn't" და არა როგორც "isn," "" , "t" .

ახლა უკვე როცა გაქვთ დაყოფილი გოკენებად. შეეცადეთ დათვალოთ სიტყვების და წინადადებების რაოდენობა. სიტყვად შეგვიძლია ჩავთვალოთ ყველაფერი რომელიც იწყება ასოთი მაგალითად appla იც სიტყვაა და A-1 იც. ხოლო იმის გასაგებად თუ რამდენი წინადადებაა შეგვიძლია გამოვიყენოთ პუნქტუაციის სიმბოლოების რაოდენობა მაგალითად (, ! ? და ა.შ) ეს მთლად ზუსტი არ იქნება მაგრამ კარგ მიახლოებას მოგვცემს. თუ ფაილია არ შეიცავს არც წინადადებას და არც სიტყვას შეგვიძლია ჩავთვალოთ რომ შეიცავს 1 სიტყვას და 1 წინადადებას რო 0 ზე გაყოფა არ მოხდეს grade ის დათვლის დროს.

სანამ სიტყვების მარცვლებზე გადავალთ ჯობია რომ რაც დაწერეთ გაგესტოთ. ამისთვის შეადარეთ ჩვენ ტესტებზე და პასუხებზე თქვენი პროგრამაც თუ იგივე პასუხებს აბრუნებს, თუ არა დაუკვირდით ღეტალებს, გაანალიზეთ რაში შეიძლება იყოს პრობლემა.

შემდეგი ამოცანაა დავთვალოთ მარცვლების რაოდენობა, მაგრამ ეს ძალიან რთული ამოცანაა ამიგომ გავამარტივოთ და დავთვალოთ სიტყვაში ხმოვნების რაოდენობა ( Y ის ჩათვლით) მაგრამ არ ჩავთვალოთ ის შემთხვევები როდესაც :

1. ხმოვნის შემდეგ ღგას ისევ ხმოვანი
2. ხმოვანი e როდესაც ის სიტყვის ბოლოში გხვდება.

მაგალითად სიტყვას program აქვს 2 ხმოვანი o და a.

სიტყვას peach აქვს ერთი რადგან ორი ხმოვანი მიყოლებით როა პირველი წესის თანახმად 1 არ ჩაითვლება.

სიტყვას deduce აქვს 2 ხმოვანი. მეორე წესის თანახმად ბოლო e არ ჩაითვლება.

ასევე გაითვალისწინეთ რომ მაგალითად სიტყვა Me ჩვენი წესების თანახმად არც ერთი ხმოვანი არ აქვს მაგრამ ეს არ იქნება სწორი ამიგომ ჩავთვალოთ რომ ნებისმიერ სიტყვას აქვს 1 მაინც ხმოვანი.

ბოლო თასქი ალბათ ყველაზე რთულია ამ ამოცანაში ამიგომ ყურადღებით დაწერეთ და რომ დაწერთ შეგიძლიათ გაგესტოთ ჩვენ მოცემულ ტესტებზე. ასევე შეგიძლიათ შემდეგ თქვენი საყვარელი წიგნები წააკითხოთ პროგრამას და გაიგოთ თუ რა ღონებზე შეაფასებს.

### შესაძლო გაგრძელებები პროგრამების (extentions)

1. მონეტების ამოცანაში შეგიძლიათ განამოგაღოთ და დაწეროთ ნებისმიერი მიმღევრობის მიღებაზე მაგალითად HTTHT შემდეგ გამოთვალოთ თუ საშუალოდ რამდენი სვლა დაჭირდა ამის მიღებას და HHHHH ამ მიმღევრობის მიღებას და ახსნთ შედეგი. მიმღევრობა შეგიძლიათ წინასწარ შეაყვანიოთ მომხმარებელს
2. პასკალის სამკუთხედს უამრავი გამოყენება აქ შეგიძლიათ ნახოთ მოცემულ ლინკზე. [https://www.youtube.com/watch?v=Yhlv5Aeuo\\_k](https://www.youtube.com/watch?v=Yhlv5Aeuo_k). extention ისთვის კი თქვენმა პროგრამამ უნდა გამოაჩინოს ეს სამკუთხედი ვიზუალზე და გვაჩვენოს მისი საოცარი თვისებები.
3. **Numeric Conversions.** ამ ამოცანაში შეგიძლიათ დაწეროთ რომაული სიმბოლოების ჩვეულებრივ რიცხვებში გადმოყვანა ან მაგალითად 16 ობითი სისტემიდან ჩვეულებრივ int ში კონვერტაცი.
4. **Flesch-Kincaid Readability:** შეგიძლიათ მოიფიქროთ უკეთესი ალგორითმი მარცვლების დასათვლელად ან წინადადების უკეთ დასათვლელად. შეგიძლიათ გამოიყენოთ ტექსტის სხვა თვისებები და შეცვალოთ ზემოთ აღწერილი ფორმულა და ნახოთ უკეთეს შედეგებს თუ მიიღებთ