Forritun Haust 2017 - Lokapróf

Fullar lýsingar á öllum dæmum eru í þessu skjali.
Code::Blocks project fyrir hvert verkefni er í möppu með sama nafni og yfirskrift verkefnisins.
Munið að sé annað ekki tekið fram er **ekki** leyfilegt að nota innbyggða klasa, svo sem vector eða aðra klasa úr C++ standard template library.
Krossaspurningar eru á hlekknum Exam á skjáborði próftölvunnar.

Full descriptions of each assignment are in this document.
Code::Blocks projects for each assignment are in folders with the same title as the asignments.
Remember that unless otherwise stated use of build in classes, such as vector or other classes from the C++ standard template library is **not** allowed.
Single Choice questions are behind the link Exam on the exam computer's desktop.

Dæmi 1 - LowestValue

Í hverjum lið af þessu dæmi er útfært nýtt fall. Í lokaútgáfu verkefnisins skal main() fallið innihalda alla liðina í röð, þannig að hver keyrist á eftir öðrum.

a) (**5%**) Útfærið fall sem skilar lægsta gildi tveggja heiltalna.
Skrifið í main() fallið forritsbút sem býður notanda upp á að slá inn tvær tölur, kallar því næst á fallið og skrifar út útkomuna úr fallinu.

b) (**5%**) Útfærið fall sem skilar lægsta gildi þriggja heiltalna.
Skrifið í main() fallið forritsbút sem býður notanda upp á að slá inn þrjár tölur, kallar þvínæst á fallið og skrifar út útkomuna úr fallinu.

c) (**10%**) Útfærið fall sem skilar lægsta gildi allra talna í kviklegu fylki af heiltölum.
Skrifið í main() fallið forritsbút sem býður notanda að slá inn hve margar tölur verða í fylkinu (það má gera ráð fyrir því að slegin verði inn tala sem er stærri en 1). Þvínæst er notandanum boðið að slá inn tölurnar hverja af annarri. Að lokum er kallað á fallið og útkoma þess skrifuð út.

Assignment 1 - LowestValue

In each part of this assignment you are to create a new function. In the final version of this assignment the main function should hold code for all the parts in that order so that each part will execute.

a) (**5%**) Implement a function that returns the lowest value of two integer values. In the main function you are to write code that prompts the user to input two numbers. Then the function that is to be implemented in this part should be called and the result of that function call should be printed.

b) (**5%**) Implement a function that returns the lowest value of three integer values. In the main function you are to write code that prompts the user to input three numbers.Then the function that is to be implemented in this part should be called and the result of that function call should be printed.

c) (**10%**) Implement a function that returns the lowest integer value from a dynamic array. In the main function you are to write code that prompts the user to input how many numbers should be in the array(you may assume that a number greater than 1 will be input). Next the user should be able to input that amount of numbers into the array. Finally the function that is to be implemented in this part should be called and the result of that function call should be printed.

Dæmi 2 - CaseSwitch

ASCII gildi stafs (character) er það tölugildi sem breyta geymir til að tákna þennan staf. *ASCII* gildi hástafa í ensku eru öll á bilinu frá og með ***65*** til og með ***90***. ASCII gildi lágstafs er alltaf ***32*** hærra en *ASCII* gildi samsvarandi hástafs. Þar af leiðir að lágstafir í ensku eru allir á bilinu frá og með ***97*** til og með ***122***.
Í *b-lið* mega nemendur nota breytu af taginu ***string***, lesa inn í hana og vísa svo í hana eins og um fylki af stöfum væri að ræða *eða* nota fylki af *char* og lesa inn í það í einni *cin* skipun. Enn fremur mega nemendur nota innbyggða klasann ***vector*** ef þeim þykir það henta betur.
Nemendum er í sjálfsvald sett hverja þessara leiða þeir nota.

Ef nemendur velja að nota char array má lesa nánar um virkni þess hér: Þegar lesið er inn í fylki (array) af stöfum (character) þá les cin skipunin þangað til kemur að bili eða línubili og setur stafgildið '\0' sem hefur tölugildið 0 fyrir aftan seinasta stafinn sem lesinn er inn. ***Því þarf fylkið að vera einum lengra heldur en fjöldi stafa í strengnum***. Þetta tölugildi 0 (stafgildi '\0') má nota til að þekkja endinn á strengnum.

a) (**10%**) Bætið í main() fallið forritsbút sem býður notanda að slá inn staf (character). Sé stafurinn hástafur skrifar forritið út samsvarandi lágstaf, sé hann lágstafur skrifar forritið út samsvarandi hástaf og sé það annað tákn en enskur bókstafur skal forritið skrifa út villuskilaboð.

b) (**10%**) Bætið í main() fallið forritsbút sem býður notanda upp á að lesa inn streng og skrifar síðan allan strenginn út, nema að búið er að breyta hverjum einasta hástaf í samsvarandi lágstaf og hverjum lágstaf í samsvarandi hástaf. Gera má ráð fyrir að strengurinn sem sleginn er inn sé aldrei lengri en 9 stafir og innihaldi engin tákn önnur en enska bókstafi. Fullgild lausn skal nota fall sem tekur við strengnum (sem fylki af stöfum, string, vector) og annaðhvort skilar annarri breytu af sama tagi þar sem stöfum hefur verið breytt, eða skilar engu og breytir stöfunum inni í breytunni sjálfri. Nemendur mega velja hvorri aðferðinni er beitt, en fallið skal ekki gera annað en að breyta strengnum.

Assignment 2 - CaseSwitch

The ASCII value of a character is the integer number that represents that given character. *ASCII* values for upper case letters in the english alphabet are in the range ***65*** - ***90 (65 and 90 included)***. ASCII values for lower case letters are always ***32*** numbers higher than the ASCII value for the corresponding upper case letter. This means that every lower case letter in the english alphabet is in the range ***97*** - ***122 (97 and 122 included)***.
In *part b* students can use a variable of the built in type ***string***, read it in and then access its characters as if it were an array *or* they can use an array of *char* and read into it in a single *cin* command. Furthermore, students can use the built in class vector if they feel it works better for them.
Students are free to choose whichever of these methods they use.

If students choose to use a char array they can read more about its functionality here: When a character array is read using cin, the input is read until the next whitespace. Then the character value '\0' which has the numerical value 0 behind the last read letter. ***The array thus needs to be one longer than the number of characters in the string.*** This numerical value 0 (character value '\0') can be used to detect the end of the string.

a) (**10%**) In this part of the assignment you are to add to the main function some code that lets the user input a single character. If the character that is input is a lowercase letter, the corresponding uppercase letter should be printed to the screen. If the character is a uppercase letter, the corresponding lowercase letter should be printed to the screen. If a character is input that is not a letter of the english alphabet a error message should be printed to the screen.

b) (**10%**) In this part of the assignment you are to add code to the main function that lets the user input a string and then prints the string again, except that each uppercase letter has been changed to lowercase and each lowercase letter has been changed to uppercase. You can assume that the string enetered is never nolger than 9 characters and that every character will be an English letter. For full marks use a function that takes the string as a parameter (as a character array, string or vector) and either returns another variable of the same type where the letters have been changed or returns nothing and changes the letters within the variable. Students can choose which method they use, but the function shall not do anything apart from changing the letters in the string.

Dæmi 3 - SuperTeam

Klasinn Superhero heldur utan um nafn, aldur og ofurkraft ofurhetju.
Gefnar eru skrárnar Superhero.h og Superhero.cpp. Þar skal útfæra klasann Superhero sem skal innihalda breytur fyrir nafn (strengur), aldur (heiltala) og ofurkraft (stafur, e. character). Sjáanlegi hluti klasans (interface/ADT) skal bjóða upp á eftirfarandi aðgerðir:

- Færibreytulaus smiður sem stillir nafnið á tóman streng (""), aldur á 0 og ofurkraft á 'n'
- Smiður með þrjár færibreytur sem taka gildi fyrir hverja klasabreytu
- Fallið set_name sem tekur inn streng og yfirskrifar gildi klasabreytunnar
- Fallið set_age sem tekur inn heiltölu og yfirskrifar gildi klasabreytunnar
- Fallið set_super_power sem tekur inn staf og yfirskrifar gildi klasabreytunnar
- << (ostream)
  Fallið skrifar út nafn hetjunnar, bil, aldur hennar innan sviga, tvípunkt, bil og að lokum ofurkraftinn á eftirfarandi hátt:
  Ef breytan superpower inniheldur ***f*** er skrifað út ***Flying***
  Ef breytan superpower inniheldur ***g*** er skrifað út ***Giant***
  Ef breytan superpower inniheldur ***h*** er skrifað út ***Hacker***
  Ef breytan superpower inniheldur ***n*** er skrifað út ***None***
  Ef breytan superpower inniheldur eitthvað annað: ***Weakling***
  Dæmi: `Bjorgvin (27): Hacker`

Í main.cpp er main() fall sem kallar á tvö föll, annað (createSomeHeroes()) er þegar útfært að fullu en hitt (createGroupOfHeroes()) er óútfært.

a) (**10%**) Útfærið klasann Superhero, þannig að fallið *createSomeHeroes()* virki rétt.
   Ekkert þarf að gera í main.cpp, einungis þarf að vinna í klasaskránum.

b) (**10%**) Útfærið fallið *CreateGroupOfHeroes()*.
   Fallið skal fyrst spyrja notandann hve margar hetjur eiga að tilheyra hópnum. Þvínæst býður það notandanum að slá inn hverja og eina hetju, fyrst nafn, svo aldur og að lokum stafinn sem táknar ofurkraftinn. Þegar allar hetjurnar hafa verið slegnar inn skrifar fallið þær allar út í þeirri röð sem þær voru slegnar inn.
   Ekki þarf að vinna í klasaskránum, bara í fallinu *createGroupOfHeroes()* í main.cpp.
   Ekki má nota neina innbyggða klasa við lausn þessa liðar!

Ef ekki tekst að útfæra a-lið að fullu má samt útfæra b-lið og mögulegt er að fá full stig þar.

Assignment 3 - SuperTeam

The class Superhero holds data for a name, age and a super power of a super hero.
The files Superhero.h and Superhero.cpp are given. You are to implement the superHero class that should have the following member variables: a name(a string), an age(an integer) and a super power(a character). The public part of the class(the interface/ADT) should have the following methods/functions:

- A constructor that takes no parameters. The default value for the name should be an empty string(""), for age it should be 0 and for the super power it should be 'n'.
- A constructor that takes 3 parameters, one for each variable of the class.
- The function setName which takes a string and applies the value to name.
- The function setAge which takes an integer as a parameter and applies the value to age.
- The function setSuperpower which takes a character as a parameter and gives the variable that represents the superpower that value.
- << (ostream)
  The function outputs the name of the hero, a space, the heros age within parenthesis followed by a colon, a space and finally the superpower by the following set of rules:
  If the variable superpower has the value ***f*** the output should be ***Flying***
  If the variable superpower has the value ***g*** the output should be ***Giant***
  If the variable superpower has the value ***h*** the output should be ***Hacker***
  If the variable superpower has the value ***n*** the output should be ***None***
  If the variable superpower has any other value the output should be ***Weakling***
  Example: `Bjorgvin (27): Hacker`

In main.cpp is a main() function that calls two function, one of the (createSomeHeroes()) is already implemented but the other one (createGroupOfHeroes()) has not been implemented.

a) (**10%**) Implement the class Superhero so the function *createSomeHeroes()* works correctly.
   Nothing needs to be done in the main function, you only need to work in the class files for this part.

b) (**10%**) implement the function *CreateGroupOfHeroes()*.
   The function should ask the user how many heroes should be in the group. Next the function should let the user input each hero one at a time, first the name of the hero, then the age and finally the letter that represents the hero´s superpower. Once all the heroes have been input the function should print the heroes to the screen in the same order as they were input. For this problem you will only need to work in the operation *createGroupOfHeroes()* within the file main.cpp.
   It is prohibited to use any built in classes in this part of the assignment!

If you can not implement part ***a*** fully you can still implement part ***b*** and get full points there.

Dæmi 4 - DoubleList

Klasinn *DoubleList* heldur utan um lista af rauntölum. Þegar klasinn er smíðaður þarf að taka fram hve margar tölur eigi að vera í listanum en eftir það er hægt að lesa inn allan listann í einu, setja inn og fá út gildi á tilteknar staðsetningar í listanum og skrifa listann út í heild sinni.
Ekki má nota neina innbyggða klasa við lausn þessa verkefnis!
Ekki þarf að breyta neinu í main(), einungis að útfæra klasann *DoubleList*.

a) (**10%**) Búið til klasann DoubleList þannig að main fallið virki (utan aðgerðina +, sem er útfærð í b-lið). Klasinn skal innihalda eftirfarandi aðgerðir:
   - Smiður sem tekur inn fjölda gilda sem listinn mun hafa pláss fyrir
   - setAt(index, value) sem yfirskrifar gildið á staðsetningunni index á value
     - Ef gildi index er fyrir utan fylkið skal fallið ekki gera neitt
   - << (ostream) sem skrifar út hvert stak í listanum með bil á milli
   - >> (istream) sem leyfir notandanum að slá inn jafnmorg stök og listinn tekur

b) (**10%**) Yfirskrifið (overload) plús *virkjann* (+) á klasanum þannig að virkinn leggi saman tvö tilvik af klasanum *DoubleList*. Þessi aðgerð skal skeyta listunum saman þannig að fyrst komi öll gildin úr listanum vinstra megin við plús virkjann og svo komi öll gildin út listanum hægra megin við virkjann og skila útkomunni í nýju tilviki af *DoubleList*.

Assignment 4 - DoubleList

The class DoubleList holds a list of real numbers. When a *DoubleList* instance is constructed an amount of numbers the instance should hold must be stated. After a DoubleList instance is constructed it should be possible to input a whole list, set and get values at specific positions in the list and output the entire list.

It is prohibited to use built in classes in this assignment!

You don´t need to change the main function. You only need to implement the class *DoubleList*.

a) (**10%**) Create the class *DoubleList* so that the given main function will work excluding the *+ operator* overloading (It should be implemented in part b). The class should have the following methods/functions:
   - A constructor that takes a number that represent the size of the list
   - setAt(index, value) which sets the given value to the given index
     - If the index is outside the array, do nothing
   - << (ostream) which outputs each element of the list separated by a space
   - >> (istream) which allows the user to input as many numbers as the size of the list

b) (**10%**) Overload the plus *operator* (+) for the class *DoubleList* so it will be possible to add two instances of *DoubleList* together. This function should concatenate the two lists so that all the elements of the *DoubleList* on the left hand side of the plus operator come first followed by all the elements from the *DoubleList* on the right hand side.