



FORRITUN

T-111-PROG

Lokapróf – forritunarhluti

Kennari: Hrafn Loftsson

Dagsetning: 12. nóvember, 2015

Tími: 09:00 – 12:50

Hjálpargögn: Engin

Nafn: _____

Kennitala: _____

Lokaprófið í námskeiðinu Forritun er tvískipt. Þessi hluti (forritun) gildir **80%** og gera má ráð fyrir að **3 klst** þurfi í þennan hluta. Allar spurningar eru settar fram á bæði íslensku og ensku. / *The final exam in the Programming course is in two parts. This part (programming) weighs 80% and you should assume 3 hours for this part. All questions are stated in both Icelandic and English.*

Stafir / Characters

Ef þú átt í vandræðum með að finna tiltekna stafi á lyklaborðinu þínu í prófinu þá eru hér nokkrir til að afrita: / *In case you have problems finding specific characters on your keyboard, here are a few to copy:*

`() {} [] < > = _ - & | / \ ! " ' , . ; : ' " < > = _ - & | / \ ! " ' , . ; : ' "`

1. Sum of even (10%)

Lýsing á íslensku

Skrifið C++ forrit sem les inn sex heiltölur af lyklaborði **inn í fylki**, reiknar út summu sléttu talnanna og skrifar summuna út. Skrifið forritið á þann veg að lesa inn færri eða fleiri tölur en sex hefur mjög litla breytingu í för með sér. Forritið skal innihalda tvö föll til viðbótar við `main()` fallið:

- *readNumbers*: les inn tölurnar af lyklaborði inn í fylki
- *sumOfEven*: skilar summu sléttu talnanna úr gefnu fylki

English description

Write a C++ program that reads six integers from the keyboard **into an array**, calculates the sum of the even numbers, and writes out the sum. Write the program in such a way that reading fewer or more than six numbers results in a minor change. The program should contain two functions in addition to the `main()` function:

- *readNumbers*: reads the numbers from the keyboard into an array
- *sumOfEven*: returns the sum of the even numbers from a given array

Dæmi 1 / Example 1 (see file tests.txt)

1 2 3 4 5 6

Sum of even numbers: 12

Dæmi 2 / Example 2 (see file tests.txt)

1 3 5 7 9 11

Sum of even numbers: 0

2. Students (25%)

Lýsing á íslensku

Skrifið forrit sem les inn upplýsingar um $n \geq 1$ nemendur og geymir þá í **vektor**. Nemandi er með nafn (strengur með engum bilum) og einkunn (kommutala). Forritið finnur og skrifar út nemandann með hæstu einkunnina (ef tveir nemendur eru með jafn háa einkunn þá telst sá fyrri hærri) og skrifar út meðaleinkunnina.

English description

Write a program that reads in information about $n \geq 1$ students and stores them in a **vector**. A student has a name (a string without any spaces), and grade (a double). The program finds and writes out the student with the highest grade (if two students have the same grade then the former one is considered higher) and writes out the average grade.

Dæmi/Example (see file tests.txt)

```
Number of students: 4
--- Reading students ---
Name: Anna
Grade: 7.8
```

```
Name: Bjarni
Grade: 6.6
```

```
Name: Sigga
Grade: 9.5
```

```
Name: Oli
Grade: 8.4
```

```
Highest student:
Name: Sigga
Grade: 9.5
```

```
Average grade: 8.075
```

3. Multiplication table (25%)

Lýsing á íslensku

Skrifið forrit sem reiknar og skrifar út margföldunartöflu sem hefur hámarksstærð 8x8. Forritið les fyrst inn stærð af lyklaborði og villutékkar hana. Ef stærðin er innan marka þá reiknar forritið út töfluna og birtir hana á **nákvæmlega** þann máta sem sést hér fyrir neðan. Í töflunni er tab stafur ("\t") á milli talna og fjöldi undirstrika er = stærð * 8.

English description

Write a program which calculates and writes out a multiplication table, whose maximum dimension is 8x8. The program first reads in a size from the keyboard and performs error checking on it. If the size is valid, the program calculates the table and displays it in the **exact** manner as shown below. The tab character ("\t") is in between the numbers in the table and the amount of underline characters is = size * 8.

Dæmi 1 / Example 1

```
Input table size: 0
Invalid size!
```

Dæmi 2 / Example 2

```
Input table size: 9
Invalid size!
```

Dæmi 3 / Example 3

```
Input table size: 3
```

	1	2	3
1	1	2	3
2	2	4	6
3	3	6	9

Dæmi 4 / Example 4

Input table size: 7

	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	4	6	8	10	12	14
3	3	6	9	12	15	18	21
4	4	8	12	16	20	24	28
5	5	10	15	20	25	30	35
6	6	12	18	24	30	36	42
7	7	14	21	28	35	42	49

4. BlackJack (40 %)

Lýsing á íslensku

Skrifið forrit sem leyfir notanda að slá inn „BlackJack“ (21) spilahönd og reikna út gildi handarinnar. Notandinn á að geta gert þetta eins oft og hann vill. Fyrst slær notandinn inn fjölda spila og síðan gildi sérhvers spils.

Ýmis atriði sem þið þurfið að fylgja og/eða hafa í huga:

- Spilahönd samanstendur af 2 til 5 spilum. Villutékka þarf inntakið sem segir til um fjölda spila. Breyting á forritinu yfir í færri eða fleiri spil þarf að vera mjög auðveld.
- Spil á að vera útfært sem klasinn `Card`, sem hefur tvær meðlimabreytur:

```
char ch;          // The char corresponding to the
                  // value of the card
int value;        // The integer value of the card
```

- Mögulegir stafir fyrir spil eru: '2', '3', '4', '5', '6', '7', '8', '9', 't', 'j', 'q', 'k', 'a'

't'=tía, 'j'=gosi, 'q'=drottning, 'k'=kóngur, 'a'=ás

- Lágspil hafa gildi sem samsvarar viðkomandi staf. Háspil ('t', 'j', 'q', 'k') hafa gildið 10. Ásinn hefur gildið 11 (eða 1; sjá fyrir neðan).
- Skor handar á alltaf að vera sem næst 21. Ef það fer yfir 21 þá er höndin "sprungin".
- Ef höndin er sprungin en er með ás þá er hægt að lækka skor handarinnar um 10 (líkja eftir því að gildi ássins fari úr 11 í 1). Ef höndin er ennþá sprunginn eftir þetta þá er hægt að gera hið sama fyrir næsta ás, o.s.frv.
- Fallið `bool isdigit(char ch)` skilar `true` ef stafurinn `ch` er tölustafur, annars `false`.
- Inntak/úttak skal vera eins og fram kemur í dæminu hér fyrir neðan.
- Verkefnið ykkar skal innihalda þrjá skrár: `card.h`, `card.cpp`, `main.cpp`

English description

Write a program that allows the user to input a BlackJack (21) hand and then computes the value of the hand. The user should be able to do this as often as he/she wishes. The user first inputs the number of cards and then the value of each card.

A few issues that you need to follow and/or keep in mind:

- A card hand consists of 2 to 5 cards. The input corresponding to the number of cards needs to be error checked. A change in the program using less or more cards should be very easy.
- A card should be implemented as the class `Card`, with two member variables:

```
char ch;           // The char corresponding to the
                  // value of the card
int value;        // The integer value of the card
```

- The possible characters representing a card are: '2', '3', '4', '5', '6', '7', '8', '9', 't', 'j', 'q', 'k', 'a'

't'=ten, 'j'=jack, 'q'=queen, 'k'=king, 'a'=ace

- The low-value cards have a value corresponding to their characters. High-value cards ('t', 'j', 'q', 'k') have the value 10. The ace has the value 11 (or 1; see below).
- The score of a hand should be as close to 21 as possible. If the score is above 21, the hand is "busted".
- If the hand is busted, but contains an ace, then it is possible to lower the score of the hand by 10 (thus simulating that the value of the ace goes from 11 to 1. If the hand is still busted, then the same can be done for the next ace, etc.
- The function `bool isdigit(char ch)` returns `true` if the character `ch` is a digit, otherwise `false`.
- Input/output should be equivalent to what is shown in the example below.
- Your project should contain three files: `card.h`, `card.cpp`, `main.cpp`

Dæmi / Example (see file tests.txt)

```
Scoring a blackjack hand
-----
Number of cards (min 2, max 5): 2
Card no. 1: k
Card no. 2: 4
The score of the hand is: 14
Score a again? y
Number of cards (min 2, max 5): 3
Card no. 1: a
Card no. 2: 7
Card no. 3: 3
The score of the hand is: 21
Score a again? y
Number of cards (min 2, max 5): 4
```

Card no. 1: q
Card no. 2: 6
Card no. 3: 4
Card no. 4: 2
The score of the hand is: 22
Busted!
Score a again? y
Number of cards (min 2, max 5): 2
Card no. 1: a
Card no. 2: k
The score of the hand is: 21
Score a again? y
Number of cards (min 2, max 5): 5
Card no. 1: j
Card no. 2: a
Card no. 3: a
Card no. 4: a
Card no. 5: a
The score of the hand is: 14
Score a again? y
Number of cards (min 2, max 5): 6
Invalid number of cards!
Score a again? y
Number of cards (min 2, max 5): 1
Invalid number of cards!
Score a again? y
Number of cards (min 2, max 5): 4
Card no. 1: t
Card no. 2: 7
Card no. 3: a
Card no. 4: 2
The score of the hand is: 20
Score a again? n