# Подготовка за изпит

Задачи за подготовка за **онлайн** практически приемен изпит.   
към курса ["Основи на програмирането" @ СофтУни](https://softuni.bg/courses/programming-basics)

## Задача 1. Церемония за Оскарите

**Линк към Judge:** <https://judge.softuni.bg/Contests/Practice/Index/1596#0>

Филмовата академия на САЩ от 1929г. ежегодно раздава награди Оскар на грандиозна церемония. Организаторите искат да разберат колко са разходите по организирането на една такава церемония. **Напишете програма, която изчислява какви разходи ще има академията по организацията на събитието, като знаете колко е наемът на залата, в която ще се проведе.**

* **Статуетки – цената им е 30% по-малка от наема на залата**
* **Кетъринг – цената му е 15% по-малка от тази на статуетките**
* **Озвучаване – цената му е 1 / 2 от цената за кетъринг**

### Вход

От конзолата се четe **1 ред**:

* **Наем за залата – цяло число в интервала [0 … 999]**

### Изход

Да се отпечата на конзолата **колко ще са разходите по организирането на церемонията. Сумата да бъде форматирана до втория знак след десетичния знак.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3500 | 9073.75 | **Наем за залата: 3500**  **Цена за статуетки: 3500 – 30% = 2450**  **Цена за** catering  **: 2450 – 15% = 2082.5**  **Цена за озвучаване: 1 / 2 от 2082.5 = 1041.25**  **Обща цена за разходите: 3500 + 2450 + 2082.5 + 1041.25 = 9073.75** |
| 5555 | 14401.34 |  |

## Задача 2. Снимачен ден

**Линк към Judge:** <https://judge.softuni.bg/Contests/Practice/Index/1699#2>

Вие сте режисьор на филма "**Програмирането е забавно**", като имате определено време за снимки. От вас се иска да напишете програма, с която ще разберете дали **снимачният ден ще ви стигне** да заснемете филма. Снимачният ден започва с **подготовка на терен**, което е **15** **процента** от времето за снимки! Филмът има определен **брой сцени**, които се заснемат за **определено време**.

### Вход

От конзолата се **четат 3 реда**:

1. Време за снимки – **цяло число** в диапазона **[0… 1440]**
2. Брой сцени – **цяло число** в диапазона **[5… 25]**
3. Времетраене на сцена – **цяло число** в диапазона **[20… 90]**

### Изход

На конзолата да се отпечата един ред:

* Ако времето за заснемане на филма **ви стигне**:

"**You managed to finish the movie on time! You have {останало време} minutes left!**"

* Ако времето **НЕ ВИ** стигне:

"**Time is up! To complete the movie you need {нужно време} minutes.**"

**Останалото време да се закръгли до най-близкото цяло число.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 120  10  11 | Time is up! To complete the movie you need 8 minutes. | Подготовката на терен **=>** 120 \* 0.15 = 18 минути.  **Време** за **заснемане** на **сцените** **=>** 10 \* 11 = 110  **Необходимо** време: 128 минути Времето за снимки е 120 => **8 минути не стигат** за снимки. |
| 60  15  3 | You managed to finish the movie on time! You have 6 minutes left! |  |

## Задача 3. Великденска екскурзия

**Линк към Judge:** <https://judge.softuni.bg/Contests/Practice/Index/1637#4>

По време на **Великденските празници**, Деси иска да отиде на почивка. В таблицата са показани **кои са дестинациите** и **каква е цената за нощувка спрямо датите, през които си е резервирала екскурзията**.

|  |  |  |  |
| --- | --- | --- | --- |
| **Дестинация** | **21-23 март** | **24-27 март** | **28-31 март** |
| **Франция** | 30 лв. | 35 лв. | 40 лв. |
| **Италия** | 28 лв. | 32 лв. | 39 лв. |
| **Германия** | 32 лв. | 37 лв. | 43 лв. |

Напишете програма, която **изчислява колко ще струва екскурзията на Деси, като знаете дестинацията, на която иска да отиде, кога си е резервирала екскурзията и за колко нощувки ще е в дадената страна.**

### Вход

Входът се чете от конзолата и се състои от **три реда**:

* **Първи ред** - **дестинация -** **текст с възможности**"**France**"**,** "**Italy**" **или** "**Germany**"
* **Втори ред - дати, през които си е резервирала екскурзията - текст с възможности** "**21-23**",   
  "**24-27**" **или** "**28-31**"
* **Трети ред - брой нощувки - цяло число в интервала [1… 100]**

### Изход

На конзолата трябва да се отпечата **един ред**:

"**Easter trip to {дестинация} : {разходи за екскурзията} leva.**"

**Разходите за екскурзията да бъдат форматирани до втората цифра след десетичния знак.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| Germany  24-27  5 | Easter trip to Germany : 185.00 leva. | Екскурзията е до Германия и е резервирала на 24-27 март => цена за нощувката = 37 лв.  Общ разход = 5 нощувки \* 37 лв. = 185 лв. |
| Italy  21-23  7 | Easter trip to Italy : 196.00 leva. | Екскурзията е до Италия и е резервирала на 21-22 март => цена за нощувката = 28 лв.  Общ разход = 7 нощувки \* 28 лв. = 196 лв. |
| France  28-31  8 | Easter trip to France : 320.00 leva. | Екскурзията е до Франция и е резервирала на 28-31 март => цена за нощувката = 40 лв.  Общ разход = 8 нощувки \* 40 лв. = 320 лв. |

## Задача 4. Великденски яйца

**Линк към Judge:** <https://judge.softuni.bg/Contests/Practice/Index/1637#8>

Предстои Великден и едно от най-вълнуващите неща е боядисването на яйца. Наличните цветове за боядисване са:

* **червено (red)**
* **оранжев (orange)**
* **син (blue)**
* **зелен (green)**

Напишете програма, която **изчислява какъв е броят на яйцата от всеки цвят и от кой цвят яйцата са най - много, като знаете общия им брой и цвета на всяко яйце.**

### Вход

От конзолата се чете **1 ред**:

* **Броят на боядисаните яйца – цяло число в интервала [1 ... 100]**

**За всяко яйце се чете:**

* + **Цветът на яйцето – текст с възможности:** "**red**"**,** "**orange**"**,** "**blue**"**,** "**green**"

### Изход

Да се отпечатат на конзолата **5 реда**:

* "Red eggs: {брой на червените яйца}"
* "Orange eggs: {брой на оранжевите яйца}"
* "Blue eggs: {брой на сините яйца}"
* "Green eggs: {брой на зелените яйца}"
* "Max eggs: {максимален брой на яйцата от цвят} -> {цвят}"

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 7  orange  blue  green  green  blue  red  green | Red eggs: 1  Orange eggs: 1  Blue eggs: 2  Green eggs: 3  Max eggs: 3 -> green | **Брой яйца: 7**  Яйце 1: оранжево -> брой оранжеви яйца: 1  Яйце 2: синьо -> брой сини яйца: 1  Яйце 3: зелено -> брой зелени яйца: 1  Яйце 4: зелено -> брой зелени яйца: 2  Яйце 5: синьо -> брой сини яйца: 2  Яйце 6: червено -> брой червени яйца: 1  Яйце 7: зелено -> брой зелени яйца: 3  **Най-много яйца: зелени -> 3 броя** |
| **Вход** | **Изход** | **Обяснения** |
| 4  blue  red  blue  orange | Red eggs: 1  Orange eggs: 1  Blue eggs: 2  Green eggs: 0  Max eggs: 2 -> blue | **Брой яйца: 4**  Яйце 1: синьо -> брой сини яйца: 1  Яйце 2: червено -> брой червени яйца: 1  Яйце 3: синьо -> брой сини яйца: 2  Яйце 4: оранжево -> брой оранжеви яйца: 1  **Най-много яйца: сини -> 2 броя** |

## Задача 5. Дартс

**Линк към Judge:** <https://judge.softuni.bg/Contests/Practice/Index/1538#7>

Вашата задача е да напишете програма, която да изчислява, дали даден играч е успял да спечели лег. (Лег се нарича единична игра на дартс)

Първоначално играчът **започва с 301 точки**. Играчът хвърля стрелата върху таблото, като за всяко улучено поле, той получава определен брой точки. Всяко поле има по три сектора: **единичен** (Single) сектор от който се взимат броят точки от полето. Двоен (Double), от него се взимат **удвоените** **точки** от полето и троен (Triple) сектор, точките от който са **умножени по 3**.

Получените точки от всеки изстрел се изваждат от началните точки, до достигане на 0.

**Забележка:** При изстрел, даващ повече точки от наличните, той се зачита за неуспешен и играчът трябва да хвърля отново, докато не уцели точки равни на оставащите или по-малки, такъв удар се счита за успешен.

**Пример:** При наличниточки 100, удар даващ повече от 100 точки, неуспешен

При налични точки 100, удар даващ по-малко или равни на 100 точки, успешен

### Вход

Първоначално се чете **един ред**:

* **Името на играча - текст**

**След това до получаване на команда "Retire" се четат многократно по два реда:**

1. **Поле – текст ("**Single**", "**Double**" или "**Triple**")**
2. **Точки – цяло число в интервала [0… 100]**

### Изход

Играта приключва при въвеждане на команда "Retire" или при изравняване на началните 301 точки към 0. На конзолата трябва да се напечата един ред:

* Ако играчът **е спечелил лега:**
  + **"{името на играча} won the leg with {успешните изстрели} shots."**
* Ако играчът **се е отказал от играта**:
  + **"{името на играча} retired after {неуспешни изстрели} unsuccessful shots."**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| Michael van Gerwen  Triple  20  Triple  19  Double  10  Single  3  Single  1  Triple  20  Triple  20  Double  20 | Michael van Gerwen won the leg with 8 shots. | Започваме със 301 точки  Първият удар е тройно 20 -> 60 <= 301  301 – 60 = 241; успешни изстрели = 1  Втори удар е тройно 19 -> 57 <= 241  241 – 57 = 184; успешни изстрели = 2  Трети удар е двойно 10 -> 20 <=184  184 – 20 = 164; успешни изстрели = 3  Четвърти удар е единично 3 -> 3 <= 164  164 – 3 = 161; успешни изстрели = 4  Пети удар е единично 1 -> 1 <= 161  161 – 1 = 160; успешни изстрели = 5  Шести удар е тройно 20 -> 60 <= 160  160 – 60 = 100; успешни изстрели = 6  Седми удар е тройно 20 -> 60 <= 100  100 – 60 = 40; успешни изстрели = 7  Осми удар е двойно 20 -> 40 <=40  40 – 40 = 0; успешни изстрели = 8 |
| Stephen Bunting  Triple  20  Triple  20  Triple  20  Triple  20  Triple  20  Triple  20  Double  7  Single  12  Double  1  Single  1 | Stephen Bunting won the leg with 6 shots. | Започваме със 301 точки  Първият удар е тройно 20 -> 60 <= 301  301 – 60 = 241; успешни изстрели = 1  .  .  .  Петият удар е тройно 20 -> 60 <= 61  61 – 60 = 1; успешни изстрели = 5  Шестият удар е тройно 20 -> 60 > 1  Неуспешни изстрели = 1  Седмият удар е двойно 7 -> 14 > 1  Неуспешни изстрели = 2  Осмият удар е единично 12 -> 12 > 1  Неуспешни изстрели = 3  Деветият удар е двойно 1 -> 2 > 1  Неуспешни изстрели = 4  Десетият удар е единично 1 -> 1 <= 1  1 – 1 = 0; успешни изстрели = 6 |
| Rob Cross  Triple  20  Triple  20  Triple  20  Triple  20  Double  20  Triple  20  Double  5  Triple  10  Double  6  Retire | Rob Cross retired after 3 unsuccessful shots. | Започваме със 301 точки  Първият удар е тройно 20 -> 60 <= 301  301 – 60 = 241; успешни изстрели = 1  …  Пети удар е двойно 20 -> 40 <= 61  61 – 40 = 21; успешни изстрели = 5  Шести удар е тройно 20 -> 60 > 21  Неуспешни изстрели 1  Седми удар е двойно 5 -> 10 <= 21  21 – 10 = 11; успешни изстрели 6  Осми удар е тройно 10 -> 30 > 11  Неуспешни изстрели 2  Девети удар е двойно 6 -> 12 > 11  Неуспешни изстрели 3  Retire -> играча се отказва, след 3 неуспешни изстрела |

## Задача 6. Великденска украса

**Линк към Judge:** <https://judge.softuni.bg/Contests/Practice/Index/1637#11>

За великденските празници, магазин започва да продава **три вида великденска украса – кошнички за яйца (basket), великденски венци (wreath) и шоколадови зайци (chocolate bunny)**. Вашата задача е да **напишете програма, която да изчислява каква сметка трябва да плати всеки един клиент на магазина,** като се има в предвид, че **всеки** клиент **закупил четен брой продукти, ще получи 20% отстъпка от крайната цена**. След като всички клиенти приключат с покупките, трябва да се отпечата **средно по колко пари е похарчил всеки човек**.

Цените на продуктите са:

* **кошничка за яйца (basket) – 1.50 лв.”**
* **великденски венец (wreath) – 3.80 лв.**
* **шоколадов заек (chocolate bunny) – 7 лв**.

### Вход

От конзолата първоначално се чете един ред:

* **Брои на клиентите в магазина – цяло число [1… 100]**
* След това за всеки един клиент на нов ред до получаване на командата "**Finish**" се чете:
  + **Покупката която клиента е избрал – текст ("basket", "wreath" или "chocolate bunny")**

### Изход

* При получаване на командата "**Finish**" да се отпечата един ред:
  + "You purchased {броя на покупките} items for {крайната цена} leva."
* Накрая, след като всички клиенти приключат с покупките, да се отпечата на един ред
  + "Average bill per client is: {средно аритметично на парите които е похарчил всеки един клиент} leva."

**Всички пари трябва да бъдат форматирани до втората цифра след десетичния знак.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 2  basket  wreath  chocolate bunny  Finish  wreath  chocolate bunny  Finish | You purchased 3 items for 12.30 leva.  You purchased 2 items for 8.64 leva.  Average bill per client is: 10.47 leva. | В магазина има 2 клиента.  Първият купува 3 артикула -> една кошница за 1.50 лв. един венец за 3.80 лв. и един шоколадов заек за 7 лв.  Неговата сметка е 1.50 + 3.80 + 7 = 12.30 лв.  Вторият клиент купува 2 прадукта  -> един вецен за 3.80 лв и един шоколадов заек за 7 лв.  Цената е 3.80 + 7 = 10.80 лв. но тъй като е закупил четен брой продукти(2), получава 20% отстъпка.  20% от 10.80 е 2.16 лв.  Сметката, която трябва да заплати е 10.80 – 2.16 = 8.64 лв.  Средната цена която двамата клиенти са платили е  (12.30 + 8.64) / 2 = 10.47 |
| 1  basket  wreath  chocolate bunny  wreath  basket  chocolate bunny  Finish | You purchased 6 items for 19.68 leva.  Average bill per client is: 19.68 leva. | В магазина има 1 клиент.  Той купува 6 артикула -> 2 кошници по 1.50 лв. 2 венеца по 3.80 лв. и 2 шоколадови заека по 7 лева.  Цената за тях е: 2 \* 1.50 + 2 \* 3.80 + 2 \* 7 = 24.60 лв.  но тъй като е закупил четен брой продукти(6) получава 20% отстъпка.  20% от 24.60 е 4.92 лв.  Сметката която трябва да заплати е 24.60 – 4.92 = 19.68 лв.  Средната цена на клиентите в магазина е 19.68 / 1 = 19.68 |