

# Упражнения: Прости проверки

Задачи за упражнение в клас и за домашно към курса „[Основи на програмирането](#)“ @ СофтУни.

## 1. Проверка за отлична оценка

Първата задача от тази тема е да се напише **конзолна програма**, която **чете оценка** (десетично число), въведена от потребителя, и отпечатва **“Excellent!”**, ако оценката е **5.50** или по-висока.

### Примерен вход и изход:

| вход | изход      | вход | изход        | вход | изход      | вход | изход        |
|------|------------|------|--------------|------|------------|------|--------------|
| 6    | Excellent! | 5    | (няма изход) | 5.50 | Excellent! | 5.49 | (няма изход) |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#0>. Сменете стратегията на judge с "JavaScript code(NodeJS)". Трябва да получите 100 точки (напълно коректно решение):

## 2. Отлична оценка или не

Следващата задача от тази тема е да се напише **конзолна програма**, която **чете оценка** (десетично число), въведена от потребителя, и отпечатва **“Excellent!”**, ако оценката е **5.50** или по-висока, или **“Not excellent.”** в противен случай.

### Примерен вход и изход:

| вход | изход      | вход | изход          | вход | изход      | вход | изход          |
|------|------------|------|----------------|------|------------|------|----------------|
| 6    | Excellent! | 5    | Not excellent. | 5.50 | Excellent! | 5.49 | Not excellent. |

Тествайте в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#1>.

## 3. Четно или нечетно

Да се напише програма, която **чете цяло число**, въведено от потребителя, и печата дали е **четно** или **нечетно**.

### Примерен вход и изход:

| вход | изход | вход | изход | вход | изход | вход | изход |
|------|-------|------|-------|------|-------|------|-------|
| 2    | even  | 3    | odd   | 25   | odd   | 1024 | even  |

Тествайте в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#2>.

## 4. Намиране на по-голямото число

Да се напише програма, която **чете две цели числа**, въведени от потребителя, и отпечатва по-голямото от двете.

### Примерен вход и изход:

| вход   | изход | вход   | изход | вход     | изход | вход    | изход |
|--------|-------|--------|-------|----------|-------|---------|-------|
| 5<br>3 | 5     | 3<br>5 | 5     | 10<br>10 | 10    | -5<br>5 | 5     |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#3>.

## 5. Изписване на число до 10 с думи

Да се напише програма, която чете **цяло число** в диапазона [0...10], въведено от потребителя, и го **изписва с думи** на английски език. Ако числото е извън диапазона, изписва "number too big".

### Примерен вход и изход:

| вход | изход |
|------|-------|
| 5    | five  |

| вход | изход |
|------|-------|
| 1    | one   |

| вход | изход |
|------|-------|
| 9    | nine  |

| вход | изход          |
|------|----------------|
| 10   | number too big |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#4>.

## 6. Бонус точки

Дадено е **цяло число** – брой точки. Върху него се начисляват **бонус точки** по правилата, описани по-долу. Да се напише програма, която пресмята **бонус точките** за това число и **общия брой точки** с бонусите.

- Ако числото е **до 100** включително, бонус точките са 5.
- Ако числото е **по-голямо от 100**, бонус точките са **20%** от числото.
- Ако числото е **по-голямо от 1000**, бонус точките са **10%** от числото.
- Допълнителни бонус точки (начисляват се отделно от предходните):
  - За **четно** число 🏆 + 1 т.
  - За число, което **завършва на 5** 🏆 + 2 т.

### Примерен вход и изход:

| вход | изход   |
|------|---------|
| 20   | 6<br>26 |

| вход | изход     |
|------|-----------|
| 175  | 37<br>212 |

| вход | изход           |
|------|-----------------|
| 2703 | 270.3<br>2973.3 |

| вход  | изход             |
|-------|-------------------|
| 15875 | 1589.5<br>17464.5 |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#5>.

## 7. Сумиране на секунди

Трима спортни състезатели финишират за някакъв **брой секунди** (между 1 и 50). Да се напише програма, която чете времената на състезателите, въведени от потребителя, и пресмята **сумарното им време** във формат "**минути:секунди**". Секундите да се изведат с **водеща нула** (2 🏆 "02", 7 🏆 "07", 35 🏆 "35").

### Примерен вход и изход:

| вход           | изход |
|----------------|-------|
| 35<br>45<br>44 | 2:04  |

| вход          | изход |
|---------------|-------|
| 22<br>7<br>34 | 1:03  |

| вход           | изход |
|----------------|-------|
| 50<br>50<br>49 | 2:29  |

| вход           | изход |
|----------------|-------|
| 14<br>12<br>10 | 0:36  |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#6>.

## 8. Конвертор за мерни единици

Да се напише програма, която **преобразува** разстояние между следните 8 мерни единици: **m, mm, cm, mi, in, km, ft, yd**. Използвайте съответствията от таблицата по-долу:

| входна единица | изходна единица           |
|----------------|---------------------------|
| 1 meter (m)    | 1000 millimeters (mm)     |
| 1 meter (m)    | 100 centimeters (cm)      |
| 1 meter (m)    | 0.000621371192 miles (mi) |
| 1 meter (m)    | 39.3700787 inches (in)    |
| 1 meter (m)    | 0.001 kilometers (km)     |
| 1 meter (m)    | 3.2808399 feet (ft)       |
| 1 meter (m)    | 1.0936133 yards (yd)      |

Входните данни се състоят от три аргумента(реда), въведени от потребителя:

- Първи аргумент: число за преобразуване
- Втори аргумент: входна мерна единица
- Трети аргумент: изходна мерна единица (за резултата)

Резултатът да се форматира до осмия знак след десетичната запетая.

### Примерен вход и изход:

| вход           | изход                | вход            | изход                | вход            | изход                 |
|----------------|----------------------|-----------------|----------------------|-----------------|-----------------------|
| 12<br>km<br>ft | 39370.07886932441 ft | 150<br>mi<br>in | 9503999.993935991 in | 450<br>yd<br>km | 0.4114799993745505 km |

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/152#7>.

## 9. Познай паролата

Да се напише програма, която **чете парола** (един ред с произволен текст), въведена от потребителя, и проверява дали въведеното **съвпада** с фразата **"s3cr3t!P@ssw0rd"**. При съвпадение да се изведе **"Welcome"**. При несъвпадение да се изведе **"Wrong password!"**.

### Примерен вход и изход:

| вход   | изход           | вход                | изход       | вход        | изход           |
|--------|-----------------|---------------------|-------------|-------------|-----------------|
| qwerty | Wrong password! | s3cr3t!<br>P@ssw0rd | Welcom<br>e | s3cr3t!p@ss | Wrong password! |

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/152#8>.

## 10. Число от 100 до 200

Да се напише програма, в която потребителят **въвежда цяло число** като аргумент и проверява дали е **под 100, между 100 и 200** или **над 200**. Да се отпечатаат съответно съобщения като в примерите по-долу:

## Примерен вход и изход:

| вход | изход         |
|------|---------------|
| 95   | Less than 100 |

| вход | изход               |
|------|---------------------|
| 120  | Between 100 and 200 |

| вход | изход            |
|------|------------------|
| 210  | Greater than 200 |

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/152#9>.

## 11. Еднакви думи

Да се напише програма, в която потребителят **въвежда две думи** като аргументи и проверява дали са еднакви. Да не се прави разлика между главни и малки думи. Да се изведе **“yes”** или **“no”**.

## Примерен вход и изход:

| вход           | изход |
|----------------|-------|
| Hello<br>Hello | yes   |

| вход               | изход |
|--------------------|-------|
| SoftUni<br>softuni | yes   |

| вход        | изход |
|-------------|-------|
| Soft<br>Uni | no    |

| вход          | изход |
|---------------|-------|
| beer<br>vodka | no    |

| вход           | изход |
|----------------|-------|
| HeLlO<br>hELLo | yes   |

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/152#10>.

## 12. Информация за скоростта

Да се напише програма, в която потребителят **въвежда скорост** (десетично число) като аргумент и отпечатва **информация за скоростта**. При скорост **до 10** (включително) отпечатайте **“slow”**. При скорост **над 10 и до 50** отпечатайте **“average”**. При скорост **над 50 и до 150** отпечатайте **“fast”**. При скорост **над 150 и до 1000** отпечатайте **“ultra fast”**. При по-висока скорост отпечатайте **“extremely fast”**.

## Примерен вход и изход:

| вход | изход |
|------|-------|
| 8    | slow  |

| вход | изход   |
|------|---------|
| 49.5 | average |

| вход | изход |
|------|-------|
| 126  | fast  |

| вход | изход         |
|------|---------------|
| 160  | ultra<br>fast |

| вход | изход             |
|------|-------------------|
| 3500 | extremely<br>fast |

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/152#11>.

## 13. Лица на фигури

Да се напише програма, в която потребителят **въвежда вида и размерите на геометрична фигура** като аргументи и пресмята лицето ѝ. Фигурите са четири вида: квадрат (**square**), правоъгълник (**rectangle**), кръг (**circle**) и триъгълник (**triangle**). Първият аргумент на функцията представлява вида на фигурата (**square, rectangle, circle** или **triangle**). Ако фигурата е **квадрат**, следващият аргумент е едно число – дължина на страната му. Ако фигурата е **правоъгълник**, следващите два аргумента са две числа – дължините на страните му. Ако фигурата е **кръг**, следващият аргумент е едно число – радиусът на кръга. Ако фигурата е **триъгълник**, следващите два аргумента са две числа – дължината на страната му и дължината на височината към нея. Резултатът да се закръгли до **3 цифри след десетичната точка**.

## Примерен вход и изход:

| вход        | изход |
|-------------|-------|
| square<br>5 | 25    |

| вход                  | изход |
|-----------------------|-------|
| rectangle<br>7<br>2.5 | 17.5  |

| вход        | изход   |
|-------------|---------|
| circle<br>6 | 113.097 |

| вход                  | изход |
|-----------------------|-------|
| triangle<br>4.5<br>20 | 45    |

## 14. Време + 15 минути

Да се напише програма, която **чете час и минути** от 24-часово денонощие, въведени от потребителя, и изчислява колко ще е **часът след 15 минути**. Резултатът да се отпечата във формат **hh:mm**. Часовете винаги са между 0 и 23, а минутите винаги са между 0 и 59. Часовете се изписват с една или две цифри. Минутите се изписват винаги с по две цифри, с **водеща нула** когато е необходимо.

### Примерен вход и изход:

| вход    | изход | вход    | изход | вход     | изход | вход     | изход | вход     | изход |
|---------|-------|---------|-------|----------|-------|----------|-------|----------|-------|
| 1<br>46 | 2:01  | 0<br>01 | 0:16  | 23<br>59 | 0:14  | 11<br>08 | 11:23 | 12<br>49 | 13:04 |

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/152#13>.

## 15. Еднакви 3 числа

Да се въведат 3 числа и да се отпечата дали са еднакви (yes / no).

### Примерен вход и изход:

| вход    | изход | вход    | изход | вход     | изход |
|---------|-------|---------|-------|----------|-------|
| [1,1,2] | no    | [1,1,1] | yes   | [1,33,2] | no    |

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/152#14>.

## 16. \*Изписване на число до 100 с думи

Да се напише програма, която превръща число [0...100] в текст: 25 ➡ "twenty five"

При невалиден вход изходът от програмата трябва да е "invalid number".

### Примерен вход и изход:

| вход | изход  | вход | изход       | вход | изход          |
|------|--------|------|-------------|------|----------------|
| 11   | eleven | 99   | ninety nine | -23  | invalid number |

Тествайте решението си в Judge системата: <https://judge.softuni.bg/Contests/Practice/Index/152#15>.

# Изпитни задачи от минали издания на курса

## 17. \*Цена за транспорт

Втора задача от междинния изпит на 6 март 2016. Тествайте решението си [тук](#).

Студент трябва да пропътува **n** километра. Той има избор измежду **три вида транспорт**:

- **Такси**. Начална такса: **0.70** лв. Дневна тарифа: **0.79** лв. / км. Нощна тарифа: **0.90** лв. / км.
- **Автобус**. Дневна / нощна тарифа: **0.09** лв. / км. Може да се използва за разстояния минимум **20** км.
- **Влак**. Дневна / нощна тарифа: **0.06** лв. / км. Може да се използва за разстояния минимум **100** км.

Напишете програма, която чете броя километри **n** и период от деня (ден или нощ), въведени от потребителя, и изчислява **цената на най-евтиния транспорт**.

### Вход

Програмата чете **два реда(аргумента)**:

- Първият ред(аргумент) съдържа числото **n** – брой километри – цяло число в интервала [1...5000].
- Вторият ред(аргумент) съдържа дума **"day"** или **"night"** – пътуване през деня или през нощта.

### Изход

Да се отпечата на конзолата **най-ниската цена** за посочения брой километри. Резултатът **да се форматира до втория знак след десетичната запетая**.

### Примерен вход и изход:

| Вход         | Изход | Обяснения  |
|--------------|-------|--|
| 5<br>day     | 4.65  | Разстоянието е под 20 км ☑ може да се ползва само <b>такси</b> . Началната такса е 0.70 лв. Понеже е през деня, тарифата е 0.79 лв. / км. С такси <b>цената</b> е: $0.70 + 5 * 0.79 = 4.65$ лв.  |
| 7<br>night   | 7.00  | Разстоянието е под 20 км ☑ може да се ползва само <b>такси</b> . Началната такса е 0.70 лв. Понеже е през нощта, тарифата е 0.90 лв. / км. С такси <b>цената</b> е: $0.70 + 7 * 0.90 = 7.00$ лв. |
| 25<br>day    | 2.25  | Разстоянието е над 20 км ☑ може да се ползва <b>автобус</b> , но не може да се ползва влак. Автобусът е най-евтиния възможен вариант. С автобус <b>цената</b> е: $25 * 0.09 = 2.25$ лв.          |
| 180<br>night | 10.80 | Разстоянието е над 100 км ☑ може да се ползва <b>влак</b> . Влакът е най-евтиният възможен вариант за пътуване. С влак <b>цената</b> е: $180 * 0.06 = 10.80$ лв.                                 |

## 18. \*Тръби в басейн

Втора задача от изпита на 26 март 2016. Тествайте решението си [тук](#).

Басейн с **обем V** има **две тръби**, от които се пълни. **Всяка тръба има определен дебит** (литрите вода минаващи през една тръба за един час). Работникът **пуска тръбите едновременно** и излиза за **N** часа. Напишете програма, която изкарва състоянието на басейна, **в момента, когато работникът се върне**.



## Вход

От конзолата се четат **четири реда(аргументи)**, въведени от потребителя:

- Първият ред(аргумент) съдържа числото **V** – **Обем на басейна в литри** – цяло число в интервала [1...10000].
- Вторият ред(аргумент) съдържа числото **P1** – **дебит на първата тръба за час** – цяло число в интервала [1...5000].
- Третият ред(аргумент) съдържа числото **P2** – **дебит на втората тръба за час** – цяло число в интервала [1...5000].
- Четвъртият ред(аргумент) съдържа числото **H** – **часовете които работникът отсъства** – число с плаваща запетая в интервала [1.0...24.00]

## Изход

Да се отпечата на конзолата **едно от двете възможни състояния**:

- До колко се е запълнил басейна и коя тръба с колко процента е допринесла. Всички проценти се свеждат до цяло число (без закръгляне).
  - **"The pool is [x]% full. Pipe 1: [y]%. Pipe 2: [z]%. "**
- Ако басейнът се е препълнил – с колко литра е прелял за даденото време, число с плаваща запетая, форматирано до първия знак след десетичната запетая.
  - **"For [x] hours the pool overflows with [y] liters."**

\* **Имайте предвид**, че поради свеждането до цяло число се губят данни и нормално **сборът на процентите** да е **99%**, а не **100%**.

## Примерен вход и изход:

| Вход                        | Изход  | Обяснения   |
|-----------------------------|--|---|
| 100<br>0<br>100<br>120<br>3 | <b>The pool is 66% full. Pipe 1: 45%. Pipe 2: 54%.</b>     | За 3 часа:<br>Първата тръба е напълнила – 300 л.<br>Втората тръба е напълнила – 360 л.<br>Общо – 660 л. < 1000 л. => 66% са запълнени<br>Първата тръба е допринесла с 45% (300 от 660 л.).<br>Втората тръба е допринесла с 54% (360 от 660 л.). |
| 100<br>100<br>100<br>2.5    | <b>For 2.5 hours the pool overflows with 400.0 liters.</b> | За 2.5 часа:<br>Първата тръба е напълнила – 250 л.<br>Втората тръба е напълнила – 250 л.<br>Общо – 500 л. > 100 л. => 400 л. са преляли.  |

## 19. \*Поспаливата котка Том

Втора задача от изпита на 24 април 2016. Тествайте решението си [тук](#).

Котката Том обича по цял ден да спи, за негово съжаление стопанинът му си играе с него винаги когато има свободно време. За да се наспи добре, **нормата за игра** на Том е **30 000 минути в година**. Времето за игра на Том **зависи от почивните дни на стопанина му**:

- Когато е на **работа**, стопанинът му си играе с него **по 63 минути на ден**.
- Когато **почива**, стопанинът му си играе с него **по 127 минути на ден**.

Напишете програма, която чете **броя почивни дни**, въведени от потребителя, и отпечатва дали **Том може да се наспи добре** и колко е **разликата от нормата** за текущата година, като приемем че **годината има 365 дни**.

**Пример:** 20 почивни дни -> работните дни са 345 ( $365 - 20 = 245$ ). Реалното време за игра е 24 275 минути ( $345 * 63 + 20 * 127$ ). Разликата от нормата е 5 725 минути ( $30\,000 - 24\,275 = 5\,725$ ) или 95 часа и 25 минути.

### Вход

Входът се състои от **едно число** – **броят почивни дни** – **цяло число** в интервала **[0...365]**

### Изход

На конзолата трябва да се отпечатат **два реда**.

- Ако времето за игра на Том **е над нормата** за текущата година:
  - На **първия ред** отпечатайте: **“Tom will run away”**
  - На **втория ред** отпечатайте разликата от нормата във формат:

**“{N} hours and {M} minutes more for play”**

- Ако времето за игра на Том **е под нормата** за текущата година:
  - На **първия ред** отпечатайте: **“Tom sleeps well”**
  - На **втория ред** отпечатайте разликата от нормата във формат:

**“{N} hours and {M} minutes less for play”**

### Примерен вход и изход:

| вход | изход   | коментари   |
|------|---|---|
| 20   | Tom sleeps well<br>95 hours and 25 minutes less for play  | Почивни дни: $20 * 127 = 2\,540$ минути игра<br>Работни дни: $365 - 20 = 345 * 63 = 21\,735$ минути игра<br>$30\,000 > 24\,274 \Rightarrow$ остават 5725 мин = 95 часа и 25 мин |
| 113  | Tom will run away<br>3 hours and 47 minutes more for play | Почивни дни: $113 * 127 = 14\,351$ минути<br>Работни дни: $365 - 113 = 252 * 63 = 15\,876$ минути<br>$30\,000 < 30\,227 \Rightarrow 227$ мин повече = 3 часа и 47 мин           |





## 20. \*Реколта

Втора задача от изпита на 17 юли 2016. Тествайте решението си [тук](#).

От лозе с площ  $X$  квадратни метри се заделя 40% от реколтата за производство на вино. От 1 кв.м лозе се изкарват  $Y$  килограма грозде. За 1 литър вино са нужни 2,5 кг. грозде. Желаното количество вино за продан е  $Z$  литра.

Напишете програма, която пресмята колко вино може да се произведе и дали това количество е достатъчно. Ако е достатъчно, остатъкът се разделя по равно между работниците на лозето.

### Вход

Входът се състои от точно 4 реда(аргумента), въведени от потребителя:

- 1ви ред(аргумент):  $X$  кв.м е лозето – цяло число в интервала  $[10 \dots 5000]$ ;
- 2ри ред(аргумент):  $Y$  грозде за един кв.м – реално число в интервала  $[0.00 \dots 10.00]$ ;
- 3ти ред(аргумент):  $Z$  нужни литри вино – цяло число в интервала  $[10 \dots 600]$ ;
- 4ти ред(аргумент): брой работници – цяло число в интервала  $[1 \dots 20]$ ;

### Изход

На конзолата трябва да се отпечата следното:

- Ако произведеното вино е по-малко от нужното:
  - о **"It will be a tough winter! More {недостигащо вино} liters wine needed."**
    - Резултатът трябва да е закръглен към по-ниско цяло число
- Ако произведеното вино е повече от нужното:
  - о **"Good harvest this year! Total wine: {общо вино} liters."**
    - Резултатът трябва да е закръглен към по-ниско цяло число
  - о **"{Оставащо вино} liters left -> {вино за 1 работник} liters per person."**
    - И двата резултата трябва да са закръглени към по-високото цяло число

### Примерен вход и изход:

| вход                        | изход  | коментари  |
|-----------------------------|--|--|
| 650<br>2<br>175<br>3        | Good harvest this year! Total wine: 208 liters.<br>33 liters left -> 11 liters per person. | Общо грозде: $650 * 2 = 1\ 300$<br>Вино = $40\% * 1\ 300 / 2,5 = 208$<br>$208 > 175$<br>$208 - 175 = 33$ л остават -> <b>11</b> л на човек           |
| 102<br>0<br>1.5<br>425<br>4 | It will be a tough winter! More 180 liters wine needed.                                    | Общо грозде: $1\ 020 * 1.5 = 1\ 530$<br>Вино = $40\% * 1\ 530 / 2,5 = 244.80$<br>$244.80 < 425$<br>$425 - 244.8 = 180.2$ -> <b>180</b> л не достигат |