

# Упражнение: Условни конструкции

Задачи за упражнение към курса "[Основи на програмирането](#)" @ СофтУни.

Тествайте решенията си в judge системата:

<https://alpha.judge.softuni.org/contests/conditional-statements-exercise/2376>

Упражнение: Условни конструкции .....	1
1. Сумиране на секунди.....	1
2. Бонус точки .....	2
3. Време + 15 минути .....	3
4. Магазин за детски играчки .....	4
5. Годзила срещу Конг .....	5
6. Световен рекорд по плуване .....	6
7. Пазаруване .....	7
8. Обедна почивка .....	8

## 1. Сумиране на секунди

Трима спортни състезатели финишират за някакъв **брой секунди** (между 1 и 50). Да се напише програма, която чете времената на състезателите в секунди, 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").

ВХОД	ИЗХОД
35 45 44	2:04

ВХОД	ИЗХОД
22 7 34	1:03

ВХОД	ИЗХОД
50 50 49	2:29

ВХОД	ИЗХОД
14 12 10	0:36

### Насоки:

1. Прочетете входните данни (**секундите на състезателите**):

```
int firstTime = int.Parse(Console.ReadLine());  
int secondTime = int.Parse(Console.ReadLine());  
int thirdTime = int.Parse(Console.ReadLine());
```

2. Създайте **нова променлива**, в която да съхранявате **сбора от секундите на тримата състезатели**:

```
int totalTime = firstTime + secondTime + thirdTime;
```

3. След като сте намерили **сбора от секундите** трябва да ги **превърнете в минути и секунди** (например, ако сборът е **85 секунди** това са **1 минута и 25 секунди**, защото **1 минута има 60 секунди**). Създайте си **две нови променливи**. В първата изчислете, **колко минути е сбора от секунди**, като **разделите сбора на 60**. Във втората променлива **изчислете секундите с помощта на деление с остатък (%)**. Използвайте **деление с остатък (%)**, за да вземете **остатъка при деление с 60**, което са останалите секунди. Например имате общ сбор от 134 секунди (2 минути и 14 секунди) **след целочисленото деление (/) на 60 ще получим 2**, а след **делението с остатък (%) ще получим оставащите секунди(14)**.

```
int minutes = totalTime / 60;
int seconds = totalTime % 60;
```

4. След като вече знаете **колко минути и секунди** е общия сбор, трябва да ги принтираме в правилния формат (**минути : секунди**), като ако секундите са **по-малки от 10** трябва да отпечатаме **0 преди секундите**, в противен случай просто си отпечатаваме **резултата в дадения формат**. За целта направете проверка (if). За принтирането може да използвате **шаблон**.

```
if (seconds < 10)
{
    Console.WriteLine($"{minutes}:0{seconds}");
}
else
{
    Console.WriteLine($"{minutes}{seconds}");
}
```

## 2. Бонус точки

Дадено е **цяло число** – начален брой точки. Върху него се начисляват **бонус точки** по правилата, описани по-долу. Да се напише програма, която пресмята **бонус точките, които получава числото и общия брой точки** (числото + бонуса).

- 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 26

вход	изход
175	37 212

вход	изход
2703	270.3 2973.3

вход	изход
15875	1589.5 17464.5

### Насоки:

- Прочетете входните данни (**числото**):

```
int number = int.Parse(Console.ReadLine());
```

- Създайте си **нова променлива от тип double**, в която ще си изчислявате **натрупаните бонус точки**, като и дадете **начална стойност 0.0**.

```
double bonus = 0.0;
```

- Направете **if-else-if конструкция** за първите **три проверки**, за да проверите големината числото и да изчислите бонуса.

```

if (number <= 100)
{
    bonus = 5;
}
else if (number > 1000)
{
    bonus = number * 0.1;
}
else
{
    bonus = number * 0.2;
}

if (number % 2 == 0)

```

4. Направете нова if-else-if конструкция, за да извършите проверките и изчислите допълнителния бонус.

- ако числото е четно към до момента натрупания бонус добавете 1
- ако завършва на 5 към бонуса добавете 2

За да проверите дали едно число е четно трябва да го разделите на 2 и ако получавате остатък при делението 0, то значи числото е четно, но ако получите остатък 1, това означава, че числото е нечетно. Например числото 34 е четно, защото  $34 / 2 = 17$  и остатъкът е 0, а числото 35 е нечетно, защото  $35 / 2 = 17$  с остатък 1.

За да проверите дали едно число завършва на 5 трябва да разделите числото на 10 и ако получите остатък при делението 5, то значи числото завършва на 5. Например числото  $245 / 10 = 24$  с остатък 5.

```

if (number % 2 == 0)
{
    bonus = bonus + 1;
}
else if (number % 10 == 5)
{
    bonus += 2;
}

```

5. Принтирайте на два реда резултатите. На първия ред натрупания бонус, а на втория крайното число, което ще намерите като съберете началния брой точки и бонуса.

```

Console.WriteLine(bonus);
Console.WriteLine(number + bonus);

```

### 3. 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 часове:минути. 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
1 46	2:01

вход	изход
0 01	0:16

вход	изход
23 59	0:14

вход	изход
11 08	11:23

вход	изход
12 49	13:04

## 4. Магазин за детски играчки

Петя има магазин за детски играчки. Тя получава голяма поръчка, която трябва да изпълни. С парите, които ще спечели иска да отиде на екскурзия.

Цени на играчките:

- Пъзел - **2.60** лв.
- Говореща кукла - **3** лв.
- Плюшено мече - **4.10** лв.
- Миньон - **8.20** лв.
- Камионче - **2** лв.

Ако поръчаните играчки са **50 или повече** магазинът прави **отстъпка 25% от общата цена**. От спечелените пари Петя трябва да даде **10% за наема** на магазина. Да се пресметне дали парите ще ѝ стигнат да отиде на екскурзия.

### Вход

От конзолата се четат **6 реда**:

1. Цена на екскурзията - реално число в интервала [1.00 ... 10000.00]
2. Брой пъзели - цяло число в интервала [0... 1000]
3. Брой говорещи кукли - цяло число в интервала [0 ... 1000]
4. Брой плюшени мечета - цяло число в интервала [0 ... 1000]
5. Брой миньони - цяло число в интервала [0 ... 1000]
6. Брой камиончета - цяло число в интервала [0 ... 1000]

### Изход

На конзолата се отпечатва:

- Ако парите са достатъчни се отпечатва:
  - "Yes! {оставащите пари} lv left."
- Ако парите НЕ са достатъчни се отпечатва:
  - "Not enough money! {недостигащите пари} lv needed."

Резултатът трябва да се форматира до втория знак след десетичната запетая.

### Примерен вход и изход

Вход	Изход	Обяснения
40.8 20 25 30 50 10	Yes! 418.20 lv left.	Сума: $20 * 2.60 + 25 * 3 + 30 * 4.10 + 50 * 8.20 + 10 * 2 = 680$ лв. Брой на играчките: $20 + 25 + 30 + 50 + 10 = 135$ $135 > 50 \Rightarrow$ 25% отстъпка; 25% от 680 = 170 лв. отстъпка Крайна цена: $680 - 170 = 510$ лв. Наем: 10% от 510 лв. = 51 лв. Печалба: $510 - 51 = 459$ лв. $459 > 40.8 \Rightarrow 459 - 40.8 = 418.20$ лв. остават
320 8 2	Not enough money! 238.73 lv needed.	Сума: $8 * 2.60 + 2 * 3 + 5 * 4.10 + 5 * 8.20 + 1 * 2 = 90.3$ лв. Брой на играчките: $8 + 2 + 5 + 5 + 1 = 21$ $21 < 50 \Rightarrow$ няма отстъпка

5		Наем: 10% от 90.3 = <b>9.03</b> лв.
5		Печалба: 90.3 – 9.03 = <b>81.27</b> лв.
1		<b>81.27 &lt; 320 =&gt; 320 – 81.27 = 238.73</b> лв. не достигат

## 5. Годзила срещу Конг

Снимките за дългоочаквания филм "Годзила срещу Конг" започват. Сценаристът Адам Уингард ви моли да **напишете програма**, която да изчисли, **дали предвидените средства са достатъчни** за снимането на филма. За снимките ще бъдат нужни **определен брой статисти, облекло** за всеки един статист и **декор**.

Известно е, че:

- Декорът за филма е на стойност **10% от бюджета**.
- При повече от **150** статиста, има отстъпка за облеклото на стойност **10%**.

### Вход

От конзолата се четат **3** реда:

Ред 1. Бюджет за филма – реално число в интервала [1.00 ... 1000000.00]

Ред 2. Брой на статистите – цяло число в интервала [1 ... 500]

Ред 3. Цена за облекло на един статист – реално число в интервала [1.00 ... 1000.00]

### Изход

На конзолата трябва да се отпечата **два** реда:

- Ако парите за декора и дрехите са **повече от бюджета**:
  - "Not enough money!"
  - "Wingard needs {парите недостигащи за филма} leva more."
- Ако парите за декора и дрехите са **по малко или равни на бюджета**:
  - "Action!"
  - "Wingard starts filming with {останалите пари} leva left."

Резултатът трябва да е **форматиран до втория знак** след десетичната запетая.

### Примерен вход и изход

Вход	Изход	Обяснения
20000 120 55.5	Action! Wingard starts filming with 11340.00 leva left.	Сума за декор: 10% от 20000 = 2000 лв. Сума за облекло: 120 * 55.5 = 6660 лв. Обща сума за филма: 2000 + 6660 = 8660 лв. 20000 – 8660 = 11340 лева остават.
15437.62 186 57.99	Action! Wingard starts filming with 4186.33 leva left.	Сума за декор: 10% от 15437.62 = 1543.762 лв. Сума за облекло: 186 * 57.99 = 10786.14 лв. Статистите са повече от 150 следователно има 10% отстъпка на облеклото. 10% от 10786.14 е 1078.614 10786.14 – 1078.614 = 9707.526 лв. за облекло Обща сума за филма: 1543.762 + 9707.526 = 11251.288 15437.62 – 11251.288 = 4186.331 лева остават
9587.88	Not enough money!	Сума за декор: 10% от 9587.88 = 958.788 лв.

222 55.68	Wingard needs 2495.77 leva more.	Сума за облекло: 11124.864 лв. Обща сума за филма: $958.788 + 11124.864 = 12083.652$ $9587.88 - 12083.652 = 2495.77$ лева не достигат
--------------	----------------------------------	---

## 6. Световен рекорд по плуване

Иван решава да подобри Световния рекорд по плуване на дълги разстояния. На конзолата се въвежда рекордът в секунди, който Иван трябва да подобри, разстоянието в метри, което трябва да преплува и времето в секунди, за което плува разстояние от 1 м. Да се напише програма, която изчислява дали се е справил със задачата, като се има предвид, че: съпротивлението на водата го забавя на всеки 15 м. с 12.5 секунди. Когато се изчислява колко пъти Иванчо ще се забави, в резултат на съпротивлението на водата, резултатът трябва да се закръгли надолу до най-близкото цяло число.

Да се изчисли времето в секунди, за което Иванчо ще преплува разстоянието и разликата спрямо Световния рекорд.

### Вход

От конзолата се четат 3 реда:

1. Рекордът в секунди – реално число в интервала [0.00 ... 100000.00]
2. Разстоянието в метри – реално число в интервала [0.00 ... 100000.00]
3. Времето в секунди, за което плува разстояние от 1 м. - реално число в интервала [0.00 ... 1000.00]

### Изход

Отпечатването на конзолата зависи от резултата:

- Ако Иван е подобрил Световния рекорд (времето му е по-малко от рекорда) отпечатваме:
  - "Yes, he succeeded! The new world record is {времето на Иван} seconds."
- Ако НЕ е подобрил рекорда (времето му е по-голямо или равно на рекорда) отпечатваме:
  - "No, he failed! He was {недостигащите секунди} seconds slower."

Резултатът трябва да се форматира до втория знак след десетичната запетая.

### Примерен вход и изход

Вход	Изход	Обяснения
10464 1500 20	No, he failed! He was 20786.00 seconds slower.	Иван трябва да преплува 1500 м.: $1500 * 20 = 30000$ сек. На всеки 15 м. към времето му се добавят 12.5 сек.: $1500 / 15 = 100 * 12.5 = 1250$ сек. Общо време: $30000 + 1250 = 31250$ сек. $10464 < 31250$ Времето, което не му е стигнало за да подобри рекорда: $31250 - 10464 = 20786$ сек.
Вход	Изход	Обяснения
55555.67 3017 5.03	Yes, he succeeded! The new world record is 17688.01 seconds.	Иван трябва да преплува 3017 м.: $3017 * 5.03 = 15175.51$ сек. На всеки 15 м. към времето му се добавят 12.5 сек.: $3017 / 15 = 201 * 12.5 = 2512.50$ сек. Общо време: $15175.51 + 2512.50 = 17688.01$ сек. Рекордът е подобрен: $55555.67 > 17688.01$

## 7. Пазаруване

Петър иска да купи **N** видеокарти, **M** процесора и **P** на брой рам памет. Ако броя на видеокартите е **по-голям** от този на процесорите получава **15% отстъпка** от крайната сметка. Важат следните цени:

- Видеокарта – **250 лв./бр.**
- Процесор – **35% от цената на закупените видеокарти/бр.**
- Рам памет – **10% от цената на закупените видеокарти/бр.**

Да се изчисли нужната сума за закупуване на материалите и да се пресметне дали бюджета ще му стигне.

### Вход

Входът се състои от четири реда:

- Бюджетът на Петър - **реално** число в интервала **[0.0...100000.0]**
- Броят видеокарти - **цяло** число в интервала **[0...100]**
- Броят процесори - **цяло** число в интервала **[0...100]**
- Броят рам памет - **цяло** число в интервала **[0...100]**

### Изход

На конзолата се отпечатва 1 ред, който трябва да изглежда по следния начин:

- Ако бюджета е достатъчен:  
**"You have {остатъчен бюджет} leva left!"**
- Ако сумата надхвърля бюджета:if  
**"Not enough money! You need {нужна сума} leva more!"**

Резултатът да се форматира до втория знак след десетичната запетая.

### Примерен вход и изход

Вход	Изход	Обяснения
900 2 1 3	You have 198.75 leva left!	Бюджет: 900 лв Сума за видеокарти: $2 * 250 = 500$ лв.  Цената за процесор: 35% от 500 = 175 лв. Сума за процесори: $1 * 175 = 175$ лв.  Цената за рам памет: 10% от 500 = 50 лв. Сума за рам памет: $3 * 50 = 150$ лв.  Обща сума: $500 + 175 + 150 = 825$ лв. Броя на видеокартите е по-голям от броя на процесорите, затова той получава 15% отстъпка от крайната цена: $825 - 15\% = 701.25$ лв. $701.25 \leq 900$ => парите са му достатъчни => остават $900 - 701.25 = 198.75$ лв.
920.45 3 1 1	Not enough money! You need 3.92 leva more!	Бюджет: 920.45 лв Сума за видеокарти: $3 * 250 = 750$ лв.  Цената за процесор: 35% от 750 = 262.50 лв. Сума за процесори: $1 * 262.50 = 262.50$ лв.  Цената за рам памет: 10% от 750 = 75 лв. Сума за рам памет: $1 * 75 = 75$ лв.

		<p>Обща сума: <math>750 + 262.50 + 75 = 1087.50</math> лв.</p> <p>Броя на видеокартите е по-голям от броя на процесорите, затова той получава 15% отстъпка от крайната цена: <math>1087.50 - 15\% = 924.37</math> лв.</p> <p><math>924.37 &gt; 920.45</math></p> <p>=&gt; парите не са му достатъчни</p> <p>=&gt; нужни са <math>924.375 - 920.45 = 3.92</math> лв.</p>
--	--	---

## 8. Обедна почивка

По време на обедната почивка искате да изгледате епизод от своя любим сериал. Вашата задача е да напишете програма, с която ще разберете дали **имате достатъчно време** да изгледате епизода. По време на почивката отделяте **време за обяд** и **време за отдих**. Времето за обяд ще бъде **1/8** от времето за почивка, а времето за отдих ще бъде **1/4** от времето за почивка.

### Вход

От конзолата се четат **3** реда:

1. **Име на сериал** – текст
2. **Продължителност на епизод** – цяло число в диапазона [10... 90]
3. **Продължителност на почивката** – цяло число в диапазона [10... 120]

### Изход

На конзолата да се изпише един ред:

- Ако времето е достатъчно да изгледате епизода:  
"You have enough time to watch {име на сериал} and left with {останало време} minutes free time."
- Ако времето не Ви е достатъчно:  
"You don't have enough time to watch {име на сериал}, you need {нужно време} more minutes."

Времето в двете изходни съобщения да се закръгли до най-близкото цяло число нагоре.

### Примерен вход и изход

Вход	Изход	Обяснения
Game of Thrones 60 96	You have enough time to watch Game of Thrones and left with 0 minutes free time.	<p>Време за обяд : <math>96 * 1/8 = 12.0</math></p> <p>Време за отдих : <math>96 * 1/4 = 24.0</math></p> <p>Останало време : <math>96 - 12 - 24 = 60</math></p> <p>Останалото време е по-голямо или равно на продължителността на епизода, следователно печатаме подходящия изход.</p>
Teen Wolf 48 60	You don't have enough time to watch Teen Wolf, you need 11 more minutes.	<p>Време за обяд : <math>60 * 1/8 = 7.5</math></p> <p>Време за отдих : <math>60 * 1/4 = 15.0</math></p> <p>Останало време : <math>60 - 7.5 - 15 = 37.5</math></p> <p>Останалото време е по-малко от продължителността на епизода, следователно печатаме подходящия изход.</p>



