1. Напишете функция, която получава две числа и връща по – голямото от тях. Направете същото и за по - малкото. Демонстрирайте работата на функциите като четете множество числа до въвеждане на 0 и изкарате най – голямото въведено и най-малкото въведено число.
2. 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. Примери:

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

1. Да се напише програма, която чете 3 числа и отпечатва дали са еднакви (yes / no).
2. 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 à + 1 т.
  + За число, което **завършва на 5** à + 2 т.

Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 20 | 6  26 | 175 | 37  212 | 2703 | 270.3  2973.3 | 15875 | 1589.5  17464.5 |

1. 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 à "02", 7 à "07", 35 à "35"). Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 35  45  44 | 2:04 | 22  7  34 | 1:03 | 50  50  49 | 2:29 | 14  12  10 | 0:36 |

1. Напишете програма, която проверява дали **точка {x, y}** се намира **вътре в правоъгълник {x1, y1} – {x2, y2}**. Входните данни се четат от конзолата и се състоят от 6 реда, въведени от потребителя: десетичните числа **x1**, **y1**, **x2**, **y2**, **x** и **y** (като се гарантира, че **x1 < x2** и **y1 < y2**). Една точка е вътрешна за даден правоъгълник, ако се намира някъде във вътрешността му или върху някоя от страните му. Отпечатайте “**Inside**” или “**Outside**”.
2. 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-D или нощ-N), въведени от потребителя, и изчислява **цената на най-евтиния транспорт**.

1. 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.**
2. 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**.

1. Дадени са n **цели числа** в интервала [**1**…**1000**]. От тях някакъв процент p1 са под 200, друг процент p2 са от 200 до 399, друг процент p3 са от 400 до 599, друг процент p4 са от 600 до 799 и останалите p5 процента са от 800 нагоре. Да се напише програма, която изчислява и отпечатва процентите p1, p2, p3, p4 и p5.
2. Да се напише програма, която прочита от конзолата **цяло число** n, въведено от потребителя, и чертае **крепост** с ширина **2 \*** n **колони** и височина n **реда** като в примерите по-долу. Лявата и дясната колона във вътрешността си са широки **n / 2**.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 3 | /^\/^\  | |  \\_/\\_/ | 4 | /^^\/^^\  | |  | |  \\_\_/\\_\_/ | 5 | /^^\\_\_/^^\  | |  | |  | \_ |  \\_\_/ \\_\_/ | 8 | /^^^^\\_\_\_\_/^^^^\  | |  | |  | |  | |  | |  | \_\_ |  \\_\_\_\_/ \\_\_\_\_/ |

1. Да се напише програма, която прочита от конзолата **цяло число** n, въведено от потребителя, и чертае **пеперуда** с ширина **2 \*** n - 1 **колони** и височина **2 \* (**n - 2) + 1 **реда** като в примерите по-долу. **Лявата** и **дясната** ѝ **част** са **широки** **n - 1**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **вход** | **изход** | **вход** | **изход** |
| 3 | \*\ /\*  @  \*/ \\* | 5 | \*\*\*\ /\*\*\*  ---\ /---  \*\*\*\ /\*\*\*  @  \*\*\*/ \\*\*\*  ---/ \---  \*\*\*/ \\*\*\* | 7 | \*\*\*\*\*\ /\*\*\*\*\*  -----\ /-----  \*\*\*\*\*\ /\*\*\*\*\*  -----\ /-----  \*\*\*\*\*\ /\*\*\*\*\*  @  \*\*\*\*\*/ \\*\*\*\*\*  -----/ \-----  \*\*\*\*\*/ \\*\*\*\*\*  -----/ \-----  \*\*\*\*\*/ \\*\*\*\*\* |