

2. في لعبة رمي إلى الهدف، يرمي الشخص عدداً من الأسمهم نحو الهدف. الدرجات بالنسبة لكلّ رمية هي عدد صحيح من 0 درجات (رمية ضائعة بشكل كامل) وحتى 10 درجات (إصابة في مركز الهدف).

ينجح الشخص في اللعبة إذا تحقق على الأقل واحده من الحالات التالية:

- في معدل العلامات، حصل على أعلى من 5.0 درجات.
- في أغلب الرميات حاز على 6 درجات فصاعداً.
- في رمية واحدة أو أكثر حاز على 10 درجات.

مثال: رمى شخص خمسة أسمهم وحاز على الدرجات: 5, 3, 9, 7, 5 . في المعدل حاز على 5.4 درجات $(\frac{5 + 3 + 9 + 7 + 5}{5})$ ، لذا فقد نجح في اللعبة.

مثال إضافي: رمى شخص ستة أسمهم وحاز على الدرجات: 8, 7, 1, 6, 0 . في أربع رميات من ضمن الستّ (أي في أغلب رمياته) حاز على 6 درجات فصاعداً، لذا فقد نجح في اللعبة.

مثال إضافي: رمى شخص ثلاثة أسمهم وحاز على الدرجات: 2, 0, 10 . في إحدى الرميات حاز على 10 درجات، لذا فقد نجح في اللعبة.

اكتبوا عملية خارجية باسم success بلغة Java أو Success بلغة C# ، تتلقى عدداً صحيحاً - num .
num يشير إلى عدد الأسمهم التي رماها الشخص نحو الهدف.

تستقبل العملية عدد الدرجات التي حاز عليها الشخص بالنسبة لكلّ سهم رماه.
تُعيد العملية true إذا نجح الشخص في اللعبة، خلاف ذلك تُعيد العملية false .

3

عدد موجب – num يُسمى "عدد نرجسيًا" إذا تحقق الشرط الذي أمامكم :

يرفعون كلّ واحد من أرقام num بأسّ عدد الأرقام الموجودة في num (طول العدد num)،
يجمعون كلّ النتائج، والمجموع الناتج مساوٍ لـ num .

مثال لـ "عدد نرجسي": العدد 407 ، لأنّه عندما نرفع كلّ واحد من أرقامه بالأسّ 3 (الذي هو عدد الأرقام في العدد)،
نحصل على 407 : $4^3 + 0^3 + 7^3 = 64 + 0 + 343 = 407$

مثال لعدد ليس "عدد نرجسي": العدد 58 ، لأنّه عندما نرفع كلّ واحد من أرقامه بالأسّ 2 (الذي هو عدد الأرقام في العدد)،
نحصل على 89 : $5^2 + 8^2 = 25 + 64 = 89$.

أ. اكتبوا عملية خارجية باسم IsNarc بلغة Java أو C# ، تتلقى عدداً من نمط صحيح – num
أكبر من 0 . تُعيد العملية true إذا كان العدد نرجسيًا، خلاف ذلك تُعيد العملية false .

ب. اكتبوا عملية خارجية باسم TheNarc بلغة Java أو C# ، تتلقى عدداً من نمط صحيح – n
أكبر من 0 . تطبع العملية جميع الأعداد النرجسية الموجودة من 1 حتى n (بما في ذلك 1 و n) .
ملاحظة: يجب استعمال العملية التي كتبتموها في البند "أ" .

6. معطاة عملية خارجية باسم digitSum بلغة Java أو DigitSum بلغة C# ، تتلقى عدداً صحيحاً - num1 وتعيد مجموع جميع الأرقام في العدد. يمكن استعمال العملية بدون تطبيقها.

مثال : بالنسبة لـ $num1 = 961$ ، تُعيد العملية 16 $(9 + 6 + 1)$.

"مجموع الأرقام العميق" لعدد ما، هو عدد مكون من رقم واحد ينتُج بالطريقة التالية :
يحسبون مجموع الأرقام مره تلو الأخرى، إلى أن ينتُج عدد مكون من رقم واحد.

أمثلة :

"مجموع الأرقام العميق" للعدد 5 هو 5 .

"مجموع الأرقام العميق" للعدد 36 هو 9 $(3+6)$.

"مجموع الأرقام العميق" للعدد 942378 هو 6 ، كما هو مفصل فيما يلي :

$$9 + 4 + 2 + 3 + 7 + 8 = 33$$

$$3 + 3 = 6$$

أ. (1) اكتبوا عملية خارجية باسم deepSum بلغة Java أو DeepSum بلغة C# ، تتلقى عدداً صحيحاً num1 ليس سالباً، وتعيد "مجموع الأرقام العميق" له.

(2) "مجموع الأرقام العميق" يمكن أن يكون فردياً (مثلاً 5 كما في المثال الأول أعلاه) أو زوجياً (مثلاً 6 كما في المثال الثالث أعلاه). هناك من يدعى أنه في المجال الذي بين 1 و 999999 توجد أعداد لها "مجموع أرقام عميق" زوجي أكثر من الأعداد التي لها "مجموع أرقام عميق" فردي.

اكتبوا عملية خارجية باسم isCorrect بلغة Java أو IsCorrect بلغة C# ، تُعيد true إذا كان الادعاء صحيحاً، خلاف ذلك تُعيد false .

ملاحظة : يجب الاستعانة بالعملية التي كتبتها في البند الفرعي "أ(1)" .

معطاة عملية خارجية أخرى، باسم digitExists بلغة Java أو DigitExists بلغة C# ، تتلقى عدداً صحيحاً - num ورقمًا - digit . تُعيد العملية true إذا كان الرقم digit يظهر في العدد num على الأقل مره واحدة، خلاف ذلك تُعيد false . يمكن استعمال العملية بدون تطبيقها.

ب. اكتبوا عملية خارجية باسم InBoth بلغة Java أو InBoth بلغة C# ، تتلقى عددين صحيحين : num1 و num2 . تُعيد العملية true إذا كان "مجموع الأرقام العميق" لـ num1 يظهر في العدد num2 ، وكذلك "مجموع الأرقام العميق" لـ num2 يظهر في العدد num1 ، خلاف ذلك تُعيد false .

مثال : بالنسبة لـ $num1 = 36$ و $num2 = 942378$ ، تُعيد العملية true ، لأنّ "مجموع الأرقام العميق" لـ num1 يظهر في العدد num2 (942378) ، و "مجموع الأرقام العميق" لـ num2 يظهر في العدد num1 (36) .

ملاحظة : يجب الاستعانة بالعملية التي كتبتها في البند الفرعي "أ(1)" .

الفصل الثاني (٣٠ درجة)

أجب عن اثنين من الأسئلة ٨-٦ (لكل سؤال - ١٥ درجة).

٦. في ماكينة أوتوماتيكية لبيع النقاش والمشروبات توجد 3 إمكانيات مشار إليها بواسطة الرموز:

'a' بالنسبة لشراء كيس نقاش بـ 2.50 شيقل.

'b' بالنسبة لشراء علبة مشروب بـ 4.50 شيقل.

'c' بالنسبة لشراء كيس نقاش + علبة مشروب بـ 6.00 شيقل.

في كل شروة يجب اختيار أحد الرموز فقط: 'a' أو 'b' أو 'c' ، والكمية المطلوبة من الرمز الذي اختياره. الكمية المطلوبة لا يمكن أن تكون أكبر من 4.

انتبه:

إذا اختير الرمز 'c' والكمية التي اختيرت هي 3 مثلاً، يحصل الزبون على 3 علب مشروب و 3 أكياس نقاش.

اكتب بلغة Java أو بلغة C# ، برنامجاً يستقبل عدد أكياس النقاش وعدد علب المشروب الموجودة في الماكينة في بداية اليوم معين.

افترض أنه في بداية اليوم لا توجد نقود في الماكينة.

يستقبل البرنامج جميع الشروط في ذلك اليوم. يستقبل بالنسبة لكل شروة أحد الرموز: 'a' ، 'b' ، 'c' ، والكمية المطلوبة.

ينتهي الاستقبال عندما يكون عدد أكياس النقاش أو عدد علب المشروب في الماكينة أصغر من 5.

يطبع البرنامج المبلغ المالي الذي تجمع في الماكينة بعد انتهاء الاستقبال، وعدد أكياس النقاش وعدد علب المشروب التي تبقّت في الماكينة.

ملاحظة: لا حاجة لفحص سلامة المدخلات.