

ليوناردو اخترع جهاز قياس المسافة (odometer) من خلال اسقاط حبات الحصى الصغيرة عند دوران عجلات العربة.

بحساب عدد الحصى الناتجة من عدد لفات العجلات وبذلك نستطيع قياس المسافة من خلال (odometer)

وكخبراء كمبيوتر قمنا باعداد برنامج للأودوميتر (odometer) لتحسين كفاءته.

مهمتك أن تقوم ببرمجة الأودوميتر (odometer) حسب الشروط المذكورة أدناه:

شبكة العملية

أولا

يتحرك الأودوميتر ضمن مربع افتراضي قياساته (256\*256) خلية وكل خلية تحتوي على الأغلب 15 حصى (Pebbles) وتعريف الخلية بتقاطع زوجين من الخطوط العمودية والأفقية (الإحداثيات) .

حيث انه كل احداثي يقع ضمن مدى (0,255)

مثلا الخلية أو الإحداثي (i,j) تكون الاحداثيات المجاوره لها التالية : (i, j + 1), (i, j - 1), (i + 1, j), (i - 1, j). فأي احداثي (خلية) تقع في الصف الأول أو العمود الأول تسمى خلية حدودية أو طرفية (border). جهاز الأودوميتر (odometer) يبدأ دائما من الخلية أو الاحداثي (0,0) من الزاوية الشمال الغربية وابتجاه الشمال (الجهة اليسرى).

الاوامر الاساسية

الأودوميتر (عداد قياس المسافة) يمكن أن يبرمج باستخدام الأوامر التالية

\* `left` اتجه 90 درجة الى اليسار أي بعكس اتجاه عقارب الساعة شريطة أن تبقى في نفس الخلية الحالية

مثال

إذا كان الأودوميتر يواجه الجنوب في البداية فإنها بهذه الخطوة ستتواجه الشرق بعد تنفيذ الأمر.

`right` اتجه 90 درجة الى اليمين باتجاه عقارب الساعة وابقى في نفس الخلية (مثال) إذا كنت تواجه الغرب اذا ستتجه الى الشمال بعد تنفيذ الامر

`move` تحرك وحدى واحدة الى مام (في الاتجاه الذي يواجه الاودوميتر) وذلك الى الخلية المجاورة واذا لم يكن هناك خلية مجاورة (مثال يعني اننا وصلنا الخلية طرفية أوحدية ) فان هذا الامر لن يكون له اي تأثير

`get`

احدف حصى واحدة من الخلية الحالية , وإذا لم يكن اي حصى فان هذا الامر لن يكون له اي تأثير

`put`

اضف حصى واحدة الى الخلية الحالية واذا كانت هذه الخلية تحتوي على 15 حصى فان هذا الامر لن يكون له اي تأثير علما بان الاودوميتر لن يخلو من الحصى

`halt`

قم بانهاء العملية أو التنفيذ

برنامج الاودميتر بنفس ترتيب المكتوبه في البرنامج العام البرنامج يجب ان يحتوي على امر واحد على الاكثر في كل سطر الاسطر الفارغة ستهمل . الرمز `<code>#</code>` .

مثال 1

النظر في البرنامج التالي لعداد المسافات. فإنه يأخذ العدد الى الخلية (0, 2)، التي تواجه الشرق. (انتبه اولا `<code>move >` `</code>` تم تجاهله، لأن عداد المسافات في الركن الشمالي الغربي التي تواجه الشمال.)

لا تأثير

الان عداد المسافات (الأودميتر) يواجه الشرق

تسميات الحدود والحصى

لتغيير تدفق البرنامج اعتمادا على الوضع الحالي، يمكنك استخدام التسميات التي تتحسس السلاسل (احرف كبيرة واحرف صغيرة) من 128 في معظم الرموز المختارة من `<code>A</code>`, ..., `<code>z</code>`, ..., `<code>a</code>`, `<code>9</code>`, ..., `<code>0</code>`, `<code>Z</code>`. يتم ادراج الأوامر الجديدة المتعلقة بالتسميات أدناه. في أوصاف كما يلي، `<code>"L"</code>` الدالة على أي تسمية صحيحة.

`<code>"L":</code>` (i.e. `<code>"L"</code>` `<code>:</code>` يوضح موقع التسمية في البرنامج `<code>"L"</code>` جميع التسميات المعلنه يجب ان تكون فريدة من نوعها , اعلان التسمية ليس له أي تأثير على عداد المسافات.

`<code>jump "L"</code>` — واصل تنفيذ الامر بالانتقال إلى خط مع التسمية دون قيد أو شرط `<code>"L"</code>`.

`<code>border "L"</code>` — واصل الانتقال الى الخط بالتسمية `<code>"L"</code>`, اذا كان عداد المسافة الأودميتر يواجه حافة حد الشبكة مثال (`<code>move</code>`) فان التعليمات لن يكون لها اي اثر والا فان عملية التنفيذ ستستمر بشكل اعتيادي او طبيعي وهذا الامر لن يكون له اي تأثير

`<code>pebble "L"</code>` واصل التنفيذ بالانتقال الى الخط المسمى `<code>"L"</code>`, اذا كانت الخلية الحالية تحتوي على الاقل حصى واحدة والا فان عملية التنفيذ ستستمر بشكل اعتيادي او طبيعي وهذا الامر لن يكون له اي تأثير

مثال 2

البرنامج التالي يحدد الحصى الاولى في الصف (0) ويتوقف هناك اذا كان لا يوجد اي حصى بالصف (0) فانه سيتوقف على الحد في نهاية الصف. سوف يستخدم مسميان هما `<code>davinci</code>` and `<code>leonardo</code>`.

الحصى الموجودة

نهاية الصف

عداد المسافات الاودميتر يبدأ بالدوران الى اليمين تبدأ الحلقة بالتسمية الموضحة `<code>leonardo:</code>` وتنتهي بالامر `<code>jump leonardo</code>` . في هذه الحلقة(loop) يقوم عداد المسافات بفحص الخلية الحالية (0, j) مع الخلية المجاورة (0, j + 1) على اعتبار انه الخلية السابقة موجودة. الامر `<code>halt</code>` ليس بالضرورة مهما هنا كون البرنامج سيتوقف على كل الاحوال.

الجملة

يتوجب عليك تقديم برنامج عداد مسافات الاودوميتر بلغة البرمجة كما هو موضح اعلاه والذي يجعل عداد المسافة يعمل كما هو متوقع. كل مهمة جزية انظر ادناه تصف سلوك عداد المسافة الواجب تحقيقه , والمحددات والحلول المقدمة يجب ان تكون مرضية والمحددات يجب ان تحتوي على الشيين التاليين

حجم البرنامج -يجب ان البرنامج قصيرا بما فيه الكفاية .وحجم البرنامج عبارة عن عدد الاوامر بداخله

وقت التنفيذ - البرنامج يجب ان ينتهي بسرعة كافية. وقت التنفيذ هو عدد الخطوات المنفذة وكل امر لوحده يتم عده كخطوة واحدة بغض النظر ما اذا كان هذا المر له تأثير او لا وعلان التسمية , والتعليقات والاسطر الفارغة لا تحسب كخطوات

في المثال رقم 1 حجم البرماج 4 وطول وقت التنفيذ 4 في المثال رقم 2 فان حجم البرماج 6 وعندما ينفذ في الشبكة بحصى واحدة في الخلية (0, 10) وقت التنفيذ 43 خطوة: `<code>right</code>` عشرة تكرارات في الحلقة (Loop) كل تكرار ياخذ 4 خطوات `<code>pebble davinci</code>; <code>border davinci</code>; <code>move</code>; <code>jump ></code> <code>leonardo</code>), واخيرا <code>pebble davinci</code> and <code>halt</code>.`

المهمة الفرعية رقم 1 [9 نقطة]

في البداية هناك  $X$  من الحصى في الخلية (0,0) و عدد  $y$  في الخلية (0, 1)، في حين أن كل الخلايا الأخرى فارغة. تذكر أن عدد الحصى الأقصى هو 15 في أي خلية. اكتب برنامج الذي ينتهي بالدوميتر في الخلية (0, 0) إذا كانت  $x \leq y$ ، وينتهي في الخلية (0, 1) في غير ذلك. (نحن لا نهتم باتجاه عداد المسافات في نهاية المطاف، ونحن أيضا لا نهتم بعدد الحصى الموجودة في نهاية على الشبكة (grid)، أو أين تقع .

حجم البرنامج  $\Rightarrow 100$  وقت التنفيذ  $\Rightarrow 1000$

المهمة الفرعية رقم 2 [12 نقطة]

نفس المهام المذكورة أعلاه ولكن عندما ينتهي البرنامج، يجب أن تحتوي الخلية (0, 0) بالضبط على عدد  $x$  من الحصى والخلية (0, 1) يجب أن تحتوي بالضبط على  $y$  من الحصى.

حجم البرنامج  $\Rightarrow 200$  وقت التنفيذ  $\Rightarrow 2000$

المهمة الفرعية رقم 3 [19 نقطة]

هناك تماما اثنان من الحصى في مكان ما في الصف 0: واحد في خلية (0,  $X$ )، والآخر في الخلية (0,  $Y$ )؛  $x$  و  $y$  مختلفة، و  $X + Y$  زوجية. اكتب برنامج الذي يترك عداد المسافات في الخلية (المعادلة من النسخة الانجليزية)، أي بالضبط في نقطة الوسط بين الخليتين المحتوية على الحصى. الحالة النهائية للشبكة ليست ذات صلة.

حجم البرنامج  $\Rightarrow 100$  وقت التنفيذ  $\Rightarrow 200000$

حجم البرنامج  $\Rightarrow 100$  وقت التنفيذ  $\Rightarrow 200000$

هناك في الغالب 15 حصى في الشبكة، لا يوجد اثنان منهم في نفس الخلية. اكتب برنامج يجمع كل منهم في الزاوية الشمالية الغربية. وبشكل أدق، إذا كان هناك  $X$  من الحصى في الشبكة عند البداية، يجب أن يكون هناك  $X$  من الحصى في نهاية الخلية (0, 0) وليس الحصى في مكان آخر.

درجة هذه المهمة الفرعية يعتمد على طول تنفيذ البرنامج المسلم وبمعنى آخر إذا كان  $L$  هو أعلى طول تنفيذ على العديد من حالات الاختبار فان درجتك ستكون :

32 نقطة اذا كانت  $L \leq 200\,000$ ;

\*  $32 - 32 \log_{10} (L / 200\,000)$  points if  $200\,000 < L < 2\,000\,000$ ;

0 نقطة اذا كانت  $L \leq 200\,000$ ;

حجم البرنامج  $\Rightarrow 200$

المهمة الفرعية رقم 5 [28 نقطة]

قد يكون هناك أي عدد من الحصى في كل خلية من الشبكة (طبعاً، بين 0 و 15). اكتب البرنامج الذي يجد الحد الأدنى، أي أن ينتهي مع عدد المسافات في خلية (i, j) بحيث يكون كل خلية أخرى تحتوي على الأقل ما تحتويه الخلية (i, j) من الحصى . بعد تشغيل البرنامج، عدد الحصى في كل خلية يجب ان يكون هو نفسه قبل تشغيل البرنامج.

درجة المهمة الفرعية هذه يعتمد على حجم البرنامج P من البرنامج المسلم بدقة أكثر تكون درجتك

28 نقطة اذا كانت  $P \leq 444$ ;

$28 - 28 \log_{10} (P / 444)$  points if  $444 < P < 4\,440$ ;

0 نقطة اذا كانت  $P \geq 4440$

وقت التنفيذ  $\Rightarrow 44400000$

تفاصيل تشغيل البرنامج

عليك تسليم ملف واحد فقط لكل مهمة فرعية، على ان يكون كتب وفقاً لقواعد النحو المحدد أعلاه. يمكن لكل ملف مسلم حجم أقصى من 5 ميجا كحد أعلى. لكل مهمة فرعية سيتم اختبار برنامجك الخاص بالادوميتير على بعض حالات الاختبار ، وسوف تتلقى بعض ردود الفعل على الموارد المستخدمة من قبل برنامجك. في حالة البرنامج ليس صحيح نحوياً فانه يصبح من المستحيل اختباره، سوف تتلقى معلومات عن خطأ لغوي محدد.

ليس من الضروري أن تحتوي على برامج العروض الخاصة بك عدد المسافات لجميع المهام الفرعية. إذا كان تقريركم الحالي لا يحتوي على برنامج لعدد المسافات الفرعية، يتم تلقائياً تضمين تقريركم الأخير لـ الفرعية، وإذا لم يكن هناك مثل هذا البرنامج، وسوف يسجل المهمة الفرعية صفر لهذا الطلب.

كالعادة، كانت النتيجة لتقديم هو مجموع الدرجات التي تم الحصول عليها في كل مهمة فرعية، والنتيجة المهمة هي الدرجة القصوى بين التقارير الإفراج اجتازت اختبار وتقديم الماضي.

المحاكي

لأغراض الاختبار بإمكانك استخدام محاكي الادوميتير الذي بإمكانك تغديته ببرنامجك العام وشبكات الادخال. برامج الادمومتر ستكتب بنفس الصيغة المستخدمة في التسليم

الشبكة الموصوفة بهذا الملف ستحتوي على 15 من الحصى 3 في الخلية (0,10) و 12 في الخلية (4,5)

