

問題

3章 プログラミング分野の問題

/ /

第1問 宝探しゲームのプログラム

次の文章を読み、空欄 **ア** ~ **カ** に入れるのに最も適当なものを、あとの解答群のうちから1つずつ選べ。(センター試験情報関係基礎2020 改)

吉野さんはロボットを操作して宝探しをするゲームを作成している。図1はゲーム画面の完成予想図である。ゲーム画面には、横yokoマス×縦tateマスで構成されたゲームボード(以降、ボードと呼ぶ)、各種情報、ロボットを操作するためのボタンが表示される。ボードのマスには宝が1つ、罠が複数隠されている。ゲームの目的は、ボード上のロボットを上下左右に1歩(=1マス)ずつ移動させ、罠を探知して避けながら、決められた操作回数以内で宝のマスに入れることである。

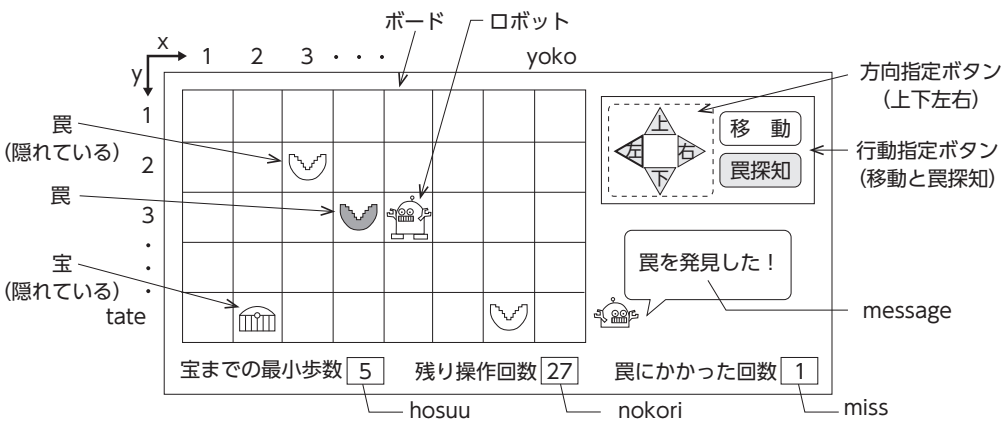


図1 ゲーム画面

ロボットと宝の位置は、マスの座標(x, y)で示す。ロボットの位置は変数 robo_x、robo_y、宝の位置は変数 takara_x、takara_y に格納する。ゲーム開始時にはロボットと宝の位置は異なるマスに配置される。

ロボットの操作は、方向指定ボタンの1つを押し、続いて行動指定ボタンを押すことで行う。行動指定ボタンには移動ボタンと罠探知ボタンがある。図2は移動ボタンが押されたときの手続きである。方向指定ボタンで方向を指定して移動ボタンを押すと、指定した方向にロボットを移動させるための変数 d_x、d_y に適切な値が代入されて図2が実行される。たとえば、下方向を指定した場合は「d_x = 0, d_y = 1」、右方向を指定した場合は「d_x = **ア**, d_y = **イ**」の代入がされて図2が実行される。

変数 nokori には残り操作回数、変数 message にはゲームの状況を示すメッセージを格納する。変数の値が更新されるとゲーム画面の数値やメッセージに反映される。図2では、ロボットのボード外への移動を防ぐ処理、残り操作回数を1減らす処理、宝のマスに入ったことを表示するための処理も行う。

第1問

- ア
- イ
- ウ
- エ
- オ
- カ

```
(01)  もし robo_x + d_x > 0 and robo_x + d_x <= ウ and
      robo_y + d_y > 0 and robo_y + d_y <= エ ならば:
(02)  |   robo_x = robo_x + d_x
(03)  |   robo_y = robo_y + d_y
(04)  nokori = nokori - 1
(05)  もし takara_x == robo_x and takara_y == robo_y ならば:
(06)  |   message = "宝を見つけた! 宝探し成功!"
```

図2 移動ボタンが押されたときのプログラム

また、ゲームのヒントとして、ロボットの位置から宝までの最少歩数を表示する。最少歩数は変数 hosuu に格納され、値が更新されるとゲーム画面の数値に反映される。図3は hosuu の計算手続きである。

```
(01)  sa_x = takara_x - robo_x, sa_y = takara_y - robo_y
(02)  もし オ ならば:
(03)  |   sa_x = sa_x * (-1)
(04)  もし カ ならば:
(05)  |   sa_y = sa_y * (-1)
(06)  hosuu = sa_x + sa_y
```

図3 宝までの最少歩数の計算プログラム

- ア** ~ **カ** の解答群
- ① 0

② -1

③ yoko

④ tate

⑤ sa_x < 0

⑥ sa_x > 0

⑦ sa_y < 0

⑧ sa_y > 0

⑨ sa_x < yoko

⑩ sa_x > yoko

㉑ sa_y < tate

㉒ sa_y > tate