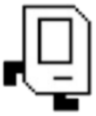
AP Computer Science A

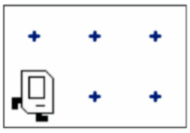
Stanford Karel[1]

Student: ____________________ Instructor: 杜博识 Date: __________________

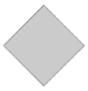
1. Karel and its world



____



____



____

____

2. Built-in Karel commands

```
move();
turnLeft();
putBeeper();
pickBeeper();
```

3. Karel program structure

```
/*
 *Comments may be included anywhere in the problem between
 *a slash-star and the corresponding star-slash characters.
 */
//This program makes Karel execute four commands in a row
function main(){
    move();
    putBeeper();
    move();
    move();
}
```

4. Method definition

```
function name(){
    //statements in the function body
}
```

```
// This function teaches Karel how to turn to the right.
function turnRight(){
   turnLeft();
   turnLeft();
   turnLeft();
}

// We can now use turnRight()
```

[1] http://stanford.edu/~cpiech/karel/lessons.html

```
function main(){
   move();
   turnRight();
   move();
   turnRight();
   move();
}
```

```
// This function teaches Karel how to turn around.
function turnAround(){
   turnLeft();
   turnLeft();
}
```

5. Newspaper

6. Repeat

```
// Put a beeper in each corner.
function main(){
   repeat(4){
      move();
      move();
      putBeeper();
      turnLeft();
   }
}
```

```
//Make karel place 50 beepers
function place50Beepers(){
   repeat(50){
      putBeeper();
   }
}

function main(){
   move();
   place50Beepers();
   move();
}
```

7. Karel condition names

frontIsClear()	frontIsBlocked()
leftIsClear()	leftIsBlocked()
rightIsClear()	rightIsBlocked()
beepersPresent()	noBeepersPresent()
beepersInBag()	noBeepersInBag()
facingNorth()	notFacingNorth()
facingEast()	notFacingEast()
facingSouth()	notFacingSouth()
facingWest()	notFacingWest()

8. while

```
while(condition){
   //statements to be executed REPEATEDLY
}
```

```
//This program will make karel pick up a pile of
//beepers no matter how big the pile is
function main(){
   move();
   while(beepersPresent()){
      pickBeeper();
   }
   move();
}
```

```
//Change this program so that Karel moves forward until
//she encounters a wall no matter how big the world is.
function main(){
   moveToWall();
}

function moveToWall(){
   while(frontIsClear()){
      move();
   }
}
```

```
//This program makes Karel pick up any and all
//beepers on the first row. It uses a while loop
//inside a while loop.
function main(){
   cleanCorner();
   while(frontIsClear()){
      move();
      cleanCorner();
   }
}

function cleanCorner(){
   while(beepersPresent()){
      pickBeeper();
   }
}
```

```
//Exercise 1 (Unit 8, Lesson 5)
//http://stanford.edu/~cpiech/karel/lessons.html#/english/unit8/lesson5

//Make Karel fill the world with beepers
function main() {













}
```

9. if

```
if(condition){
    //statements executed ONCE if condition is true
}

if(condition){
    //statements executed ONCE if condition is true
} else{
    //statements executed ONCE if condition is false
}
```

```
//Sometimes you only want to execute a block of code a
//single time if a condition passes. This program uses
//if/else statements to make karel invert beepers
function main(){
   invertBeeper();
   while(frontIsClear()){
      move();
      invertBeeper();
   }
}

function invertBeeper(){
   if(beepersPresent()){
      pickBeeper();
   } else{
      putBeeper();
   }
}
```

```
//Exercise 2 (Unit 9, Lesson 2)
//http://stanford.edu/~cpiech/karel/lessons.html#/english/unit9/lesson2

//Karel must help rebuild broken columns. Make a
//column of beepers above each beeper you find on the first row.
function main() {



}
```

```
//Exercise 3: Solving a maze
//http://stanford.edu/~cpiech/karel/ide.html , then choose World Maze

//The exit to the maze is marked by a beeper, so that Karel’s job is to navigate
//the corridors of the maze until it finds the beeper indicating the exit.
function main() {


}
```