



BRANCHE	SECTION(S)	ÉPREUVE ÉCRITE
Informatique	B	Durée de l'épreuve : 3h Date de l'épreuve :

Question 1 (10 points)

(1) 5 points

```
class Stack:

    def __init__(self):
        self.nb = 0
        self.items = []

    def isEmpty(self):
        return self.nb == 0

    def push(self, k):
        self.items.append(k)
        self.nb += 1

    def pop(self):
        if not self.isEmpty():
            self.nb -= 1
            return self.items.pop()
```

(2) 5 points

```
def reverse_order(s):
    letters = Stack()
    for letter in s:
        letters.push(letter)
    result = ''
    while not letters.isEmpty():
        result += letters.pop()
    return result
```

```
print(reverse_order('python'))
```

Question 2 (10 points)

(1) 3 points

```
from random import randint

def random_matrix(rows, cols, maxi):
    return [[randint(0, maxi) for i in range(cols)] for j in range(rows)]
```

(2) 1 point

```
def display(m):
    for row in m:
        print(row)
```

(3) 5 points

```
def s(m, x, y):
    if x == 0 and y == 0:
        return m[0][0]
    if x == 0:
        return m[0][y] + s(m, 0, y - 1)
    if y == 0:
        return m[x][0] + s(m, x - 1, 0)
    return m[x][y] + max(s(m, x - 1, y), s(m, x, y - 1))
```

(4) 1 point

```
my_matrix = random_matrix(4, 6, 10)
display(my_matrix)
print(s(my_matrix, 3, 5))
```

Question 3 (40 points)

(1) 3 points

```
import pygame, sys
from pygame.locals import *
from random import randint
from math import sqrt
```

```
pygame.init()
SIZE = 400
FPS = 60
screen = pygame.display.set_mode((SIZE, SIZE))
pygame.display.set_caption("Bouncing Ball")
clock = pygame.time.Clock()
```

(2) 2 points

```
def distance(x1, y1, x2, y2):
    return sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)
```

(3) 4 points

```
class Paddle:
    def __init__(self, x, y, width, height=10):
        self.x = x
        self.y = y
        self.width = width
        self.height = height

    def draw(self, screen):
        pygame.draw.rect(screen, Color("blue"), (self.x, self.y, self.width,
self.height))

    def move(self, dx):
        self.x += dx
```

(4a) 4 points

```
class Ball:
    def __init__(self, x, y, radius, speed=4):
        self.x, self.y = x, y
        self.radius = radius
        self.speed = speed
        self.score = 0
        self.alive = True
        self.xspeed = randint(-self.speed + 1, self.speed - 1)
        self.yspeed = -sqrt(self.speed ** 2 - self.xspeed ** 2)
```

(4b) 2+5+9 points

```
def draw(self, screen):
    x, y, r = self.x, self.y, self.radius
    pygame.draw.ellipse(screen, Color("red"), (x - r, y - r, 2 * r, 2 * r))
```

```

# pygame.draw.circle(screen, Color("red"), (x, round(y)), r)

def hit_paddle(self, paddle):
    test1 = paddle.x <= self.x < paddle.x + paddle.width and paddle.y <=
self.y + self.radius
    test2 = distance(self.x, self.y, paddle.x, paddle.y) < self.radius
    test3 = distance(self.x, self.y, paddle.x + paddle.width - 1, paddle.y)
< self.radius
    return test1 or test2 or test3

def move(self, paddle):
    if self.x + self.radius >= SIZE:
        self.xspeed = -abs(self.xspeed)
    elif self.x - self.radius < 0:
        self.xspeed = abs(self.xspeed)
    if self.y - self.radius < 0:
        self.yspeed = abs(self.yspeed)
    elif self.y >= paddle.y + paddle.height:
        self.alive = False
    if self.hit_paddle(paddle):
        self.score += 1
        self.xspeed = randint(-self.speed + 1, self.speed - 1)
        self.yspeed = -sqrt(self.speed ** 2 - self.xspeed ** 2)
        if self.score % 5 == 0:
            self.speed += 1 # (7) 2 points
    if self.alive:
        self.x += self.xspeed
        self.y += self.yspeed

# (5) 8 points
paddle = Paddle((SIZE - 60) // 2, SIZE - 30, 60)
ball = Ball(SIZE // 2, SIZE // 2, 10)
key_pressed = KMOD_NONE
done = False
while not done and ball.alive:
    for event in pygame.event.get():
        if event.type == QUIT:
            done = True
        elif event.type == pygame.KEYDOWN:
            key_pressed = event.key
        elif event.type == KEYUP:
            key_pressed = KMOD_NONE
    if key_pressed == K_RIGHT:
        paddle.move(5)
    elif key_pressed == K_LEFT:
        paddle.move(-5)
    screen.fill(Color("white"))
    paddle.draw(screen)
    ball.move(paddle)
    ball.draw(screen)
    pygame.display.update()
    clock.tick(FPS)

# (6) 1 point
print('SCORE :', ball.score)
print('GAME OVER')

pygame.quit()
sys.exit()

```