

```

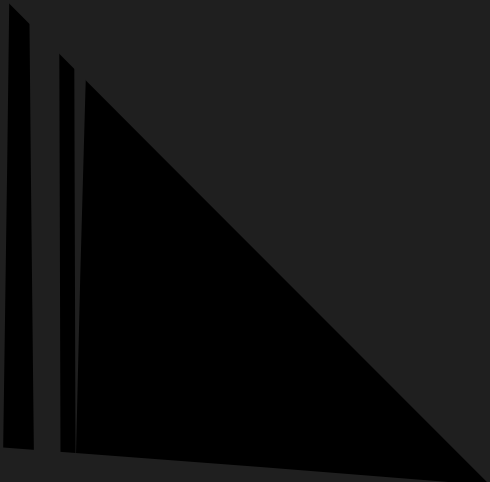
2
3 def is_prime_number(n):
4     if n < 2:
5         return False
6     elif n == 2:
7         return True
8     if n % 2 == 0:
9         return False
10
11     # Kiểm tra đến căn bậc hai của n
12     for i in range(3, int(math.sqrt(n)) + 1, 2):
13         if n % i == 0:
14             return False

```

```

1 num = int(input("Nhập số cần kiểm tra: "))
2 re ult = "là số nguyên tố" if is_prime_number(num) else "không là số n
3

```



1000

1

=== MÁY TÍNH ===

- 1. Cộng
- 2. Trừ
- 3. Nhân
- 4. Chia
- 0. Thoát

Chọn phép toán (0-4): 2

Nhập số thứ nhất: 10

Nhập số thứ hai: 11

Kết quả: $10.0 - 11.0 = -1.0$

=== MÁY TÍNH ===

- 1. Cộng
- 2. Trừ
- 3. Nhân
- 4. Chia
- 0. Thoát

Chọn phép t

Trong , người chơi đoán đến khi đúng, có gợi ý và đếm

đơn giản cách

- Sử dụng random
- Vòng lặp while cho đến khi đúng
- Mỗi lần đoán, so sánh và đưa ra gợi ý "lớn hơn" hoặc "nhỏ hơn".

```
1 import random
2
3 print("Đoán số từ 1 đến 100")
4
5 # Sinh ngẫu nhiên số bí mật từ 1 đến 100
6 so_bi_mat = random.randint(1, 100)
7 so_lan_doan = 0
8
9
10
11
12
13
14         _mat:
15             print("Số thực tế lớn hơn.")
16     elif doan > so_bi_mat:
17         print("Số thực tế nhỏ hơn.")
18     else:
19         print(f"Đúng! Số cần đoán là {so_bi_mat}. Bạn đã đoán {so_lan_doan} lần.")
20         break
21
22 except ValueError:
23     print("Vui lòng nhập một số nguyên hợp lệ!")
```

```
Đoán số từ 1 đến 100
Nhập số bạn đoán: 50
Số thực tế lớn hơn.
Nhập số bạn đoán: 75
Số thực tế nhỏ hơn.
Nhập số bạn đoán: 62
Số thực tế lớn hơn.
Nhập số bạn đoán: 65
Số thực tế lớn hơn.
Nhập số bạn đoán: 70
Số thực tế lớn hơn.
Nhập số bạn đoán: 73
Số thực tế nhỏ hơn.
Nhập số bạn đoán: 72
Số thực tế nhỏ hơn.
Nhập số bạn đoán: 71
Đúng! Số cần đoán là 71. Bạn đã đoán 8 lần.
```

```
1 def factorial_for_loop(num):
2     result = 1
3     # Kiểm tra num là số nguyên và >= 0
4     if isinstance(num, int) and num >= 0:
5         # Tính giai thừa
6         for i in range(1, num+1):
```

7
8
9
10
11
12
13
14
15
16
17
18
~~19~~


```
4     def xep_loai(diem):
5         if diem >= 8.5:
6             return "Giỏi"
7         elif diem >= 7:
8             return "Khá"
9         elif diem >= 5:
10            return "Trung bình"
11        else:
12            return "Yếu"
13
14    while True:
15        print("\n=== HỆ THỐNG QUẢN LÝ ĐIỂM ===")
16        print("1. Nhập thông tin sinh viên")
17        print("2. Hiển thị bảng điểm")
18        print("3. Tìm sinh viên điểm cao/thấp nhất")
19        print("4. Thống kê theo xếp loại")
20        print("0. Thoát")
21
22        lua_chon = input("Chọn chức năng: ")
23
24        if lua_chon == '0':
25            print("Đã thoát chương trình.")
26            break
27
28        elif lua_chon == '1':
29            n = int(input("Nhập số sinh viên muốn thêm: "))
30            for _ in range(n):
31                ten = input("Tên sinh viên: ")
32                while True:
33                    try:
34                        diem = float(input("Điểm: "))
35                        if 0 <= diem <= 10:
36                            break
37                        else:
38                            print("Điểm phải từ 0 đến 10.")
39                    except ValueError:
40                        print("Vui lòng nhập số hợp lệ.")
41                sinh_vien[ten] = diem
42                print("Đã thêm thông tin sinh viên thành công.")
43
44        elif lua_chon == '2':
45            if not sinh_v
```


2. Hiện thị bảng điểm
3. Tìm sinh viên điểm

sinh viên điểm thấp nhất (4.0): Nguyen C

=== HỆ THỐNG QUẢN LÝ ĐIỂM ===

1. Nhập thông tin sinh viên
 2. Hiện thị bảng điểm
 3. Tìm sinh viên điểm cao/thấp nhất
 4. Thống kê theo xếp loại
 0. Thoát
- Chọn chức năng: 4

Thống kê theo xếp loại:

Giỏi: 2 sinh viên

Khá: 1 sinh viên

Trung bình: 1 sinh viên

Yếu: 1 sinh viên

== HỆ THỐNG QUẢN LÝ ĐIỂM ==

1. Nhập thông tin sinh viên
 2. Hiện thị bảng điểm
 3. Tìm sinh viên điểm cao/thấp nhất
 4. Thống kê theo xếp loại
 0. Thoát
- Chọn chức năng: 0
Đã thoát chương trình.

✓ Bài 5 – Game đoán số nâng cao

Yêu cầu: Tạo game đoán số với các tính năng:

- Nhiều mức độ khó khác nhau
- Hệ thống gợi ý thông minh
- Thống kê số lần chơi và tỷ lệ thắng
- Lưu high score

```
1 import random
2
3 def game():
4     highscore = 0
5     total_games = 0
6     wins = 0
7
8     while True:
9         print("\n=== GAME ĐOÁN SỐ NÂNG CAO ===")
10        print("1. Dễ (1-10, 5 lượt)")
11        print("2. Trung bình (1-50, 7 lượt)")
```



```
63         total_games += 1
64         print(f"\nTỷ lệ thắng: {wins}/{total_games} ({wins/total_games*100:.1f}")
65         print(f"High score: {highscore}")
```

```
1 # Chạy game
2 game()
```

Đã chọn số từ 1 đến 50. Bạn có 7 lượt đoán.

Nhập dự đoán của bạn: 25

Số bạn đoán lớn hơn số đúng.

Nhập dự đoán của bạn: 12

Số bạn đoán lớn hơn số đúng.

Nhập dự đoán của bạn: 6

Số bạn đoán nhỏ hơn số đúng.

Nhập dự đoán của bạn: 9

Số bạn đoán lớn hơn số đúng.

Nhập dự đoán của bạn: 7

Số bạn đoán nhỏ hơn số đúng.

Nhập dự đoán của bạn: 8

Chính xác! Bạn đoán đúng sau 6 lượt.

New High score: 100!

Tỷ lệ thắng: 2/2 (100.0%)

High score: 100

=== GAME ĐOÁN SỐ NÂNG CAO ===

1. Dễ (1-10, 5 lượt)

2. Trung bình (1-50, 7 lượt)

Tỷ lệ thành: 2/3 (66.7%)
High score: 100

=== GAME ĐOÁN SỐ NÂNG CAO ===

1. Dễ (1-10, 5 lượt)
 2. Trung bình (1-50, 7 lượt)
 3. Khó (1-100, 10 lượt)
 0. Thoát game
- Chọn mức độ: 0
Cảm ơn bạn đã chơi!

✓ Bài 6 – Thuật toán sắp xếp với visualization

Yêu cầu: Implement các thuật toán sắp xếp và hiển thị quá trình:

Gợi ý cách làm:

- Viết hàm bubble sort, selection sort, insertion sort.
- In ra mảng sau mỗi bước sắp xếp.
- So sánh số bước, thời gian thực hiện.

```
1 import time
2
3 # ===== Các thuật toán sắp xếp =====
4
5
6
7
8
9
10
11
12
13
14         a[j + 1], a[j] = a[j], a[j + 1]
15         print(f"Bước {steps}: {a}")
16     return a, steps
17
18
19 def selection_sort(arr):
20     a = arr.copy()
21     n = len(a)
22     steps = 0
23     print("\n=== Selection Sort ===")
24     for i in range(n):
25         min_idx = i
26         for j in range(i + 1, n):
27             steps += 1
28             if a[j] < a[min_idx]:
29                 min_idx = j
```



Mảng ban đầu

```
      : [3, 4, 5, 2, 8]
Bước 7: [3, 4, 2, 5, 8]
Bước 8: [3, 4, 2, 5, 8]
Bước 9: [3, 2, 4, 5, 8]
Bước 10: [2, 3, 4, 5, 8]
```

=== Selection Sort ===

```
Bước 1: [2, 3, 8, 4, 5]
Bước 2: [2, 3, 8, 4, 5]
Bước 3: [2, 3, 4, 8, 5]
Bước 4: [2, 3, 4, 5, 8]
Bước 5: [2, 3, 4, 5, 8]
```

=== Insertion Sort ===

```
Bước 1: [5, 5, 8, 4, 2]
Sau khi chèn 3: [3, 5, 8, 4, 2]
Sau khi chèn 8: [3, 5, 8, 4, 2]
Bước 2: [3, 5, 8, 8, 2]
Bước 3: [3, 5, 5, 8, 2]
Sau khi chèn 4: [3, 4, 5, 8, 2]
Bước 4: [3, 4, 5, 8, 8]
```

khi chèn 2: [2, 3, 4, 5, 8]

=== So sánh hiệu suất ===

Bubble Sort

.000082 giây

✓ Bài 7 – Ứng dụng for/while: In mẫu ký tự / Pattern Print

Yêu cầu: Viết chương trình ứng dụng vòng lặp for, while để in ra các mẫu ký tự trên màn hình console.

- Mỗi mẫu p
lễ hợp lý.
- Cần tổ chức code thành các hàm riêng cho từng mẫu.
- Tạo menu để chọn mẫu, nhập tham số, in kết quả.

Chọn ít nhất 3 kiểu mẫu sau:

1. Hình vuông, hình chữ nhật

3.

4.

đi ý c

- Nhậ
- Dùng
- Kết h


```
24         tam_giac_vuong(n, char)
25     elif chon == "3":
26         tam_giac_can(n, char)
27     elif chon == "4":
28         tam_giac_nguoc(n, char)
29 else:
30     pr
```

TTTTTTTTT

*

Chọn một kiểu hình để in:

1. Hình vuông, hình chữ nhật
2. Tam giác vuông
3. Tam giác cân
4. Tam giác ngược
0. Thoát

Nhập lựa chọn: 0

Thoát chương trình.