NUMBER GUESSING GAME

Description :

1. The computer picks a random number between 1 and 50.
2. The user has 10 attempts to guess the correct number.
3. Feedback is provided based on how close the guess is (e.g., “Too high”, ”Too low”, “Almost there”)
4. The game exits if the guess is out of 1-50 range.
5. Remaining attempts are displayed after each guess.

```
import random
comp_input=random.randint(1,50)
rem_attempt=10
for i in range(1,11):
    user_input=int(input("Try to guess the number between 1 to 50: "))
    if user_input > 50 or user_input < 1:
      print("\nEntered input must be in between 1 to 50 !!")
      exit(0)
    rem_attempt-=1
    if user_input == comp_input :
        print("\nHurray! You guessed it right in %d moves"%(i))
        break
    if user_input > comp_input :
        if (user_input - comp_input) >10 :
            print("Too high!")
        if (user_input - comp_input) <10 :
            if (user_input - comp_input) <3 :
                print("Almost there!")
            else :
                print("High!")
    if user_input < comp_input :
        if (comp_input - user_input) >10:
            print("Too low!")
        if (comp_input - user_input) <10:
            if (comp_input - user_input) <3 :
                print("Almost there!")
            else :
                print("Low!")
    print("Remaining attempts :%d\n "%(rem_attempt))
```

```
Try to guess the number between 1 to 50: 50
Too high!
Remaining attempts :9

Try to guess the number between 1 to 50: 45
Too high!
Remaining attempts :8

Try to guess the number between 1 to 50: 40
Too high!
Remaining attempts :7

Try to guess the number between 1 to 50: 32
Too high!
Remaining attempts :6

Try to guess the number between 1 to 50: 10
High!
Remaining attempts :5

Try to guess the number between 1 to 50: 4
Almost there!
Remaining attempts :4

Try to guess the number between 1 to 50: 3
Low!
Remaining attempts :3

Try to guess the number between 1 to 50: 5
Almost there!
Remaining attempts :2

Try to guess the number between 1 to 50: 6

Hurray! You guessed it right in 9 moves
```

TO DO LIST APPLICATION

Description :

1. This is a simple task manager program that allows users to manage tasks saved in a CSV file.
2. Users can add, edit, delete and mark a particular task as complete.
3. Tasks are stored with a title, description, and status (incomplete or complete)
4. The program loads tasks from a file at startup and saves changes to the file.
5. Users can view the list of tasks and choose different tasks from a menu.

```python
import csv

FILENAME = 'tasks.csv'

def load_tasks():
    tasks = []
    try:
        with open(FILENAME, 'r') as file:
            reader = csv.DictReader(file)
            for row in reader:
                tasks.append(row)
    except FileNotFoundError:
        pass
    return tasks

def save_tasks(tasks):
    with open(FILENAME, 'w', newline='') as file:
        header = ['title', 'description', 'status']
        writer = csv.DictWriter(file, fieldnames=header)
        writer.writeheader()
        writer.writerows(tasks)

def add_task(tasks, title, description=""):
    tasks.append({'title': title, 'description': description, 'status': 'incomplete'})
    save_tasks(tasks)

def edit_task(tasks, index, title=None, description=None):
    if index >= 0 and index < len(tasks):
        if title:
            tasks[index]['title'] = title
        if description:
            tasks[index]['description'] = description
        save_tasks(tasks)

def delete_task(tasks, index):
    if index >= 0 and index < len(tasks):
        del tasks[index]
        save_tasks(tasks)

def mark_task_complete(tasks, index):
    if index >= 0 and index < len(tasks):
        tasks[index]['status'] = 'COMPLETE'
        save_tasks(tasks)

def display_tasks(tasks):
    index = 1
    for task in tasks:
        print(f"{index}. {task['title']} - {task['description']} [{task['status']}]")
        index += 1

def main():
    tasks = load_tasks()

    while True:
        print("\nTask Manager")
        print("1. Add Task")
        print("2. Edit Task")
        print("3. Delete Task")
        print("4. Mark Task Complete")
        print("5. View Tasks")
        print("6. Exit")

        choice = input("Choose an option: ")

        if choice == '1':
            title = input("Task title: ")
            description = input("Task description: ")
```

```
            add_task(tasks, title, description)
        elif choice == '2':
            index = int(input("Task number to edit: ")) - 1
            title = input("New title: ")
            description = input("New description: ")
            edit_task(tasks, index, title, description)
        elif choice == '3':
            index = int(input("Task number to delete: ")) - 1
            delete_task(tasks, index)
        elif choice == '4':
            index = int(input("Task number to mark complete: ")) - 1
            mark_task_complete(tasks, index)
        elif choice == '5':
            display_tasks(tasks)
        elif choice == '6':
            break
        else:
            print("Invalid option. Please try again.")

print(main())
```

```
Task Manager
1. Add Task
2. Edit Task
3. Delete Task
4. Mark Task Complete
5. View Tasks
6. Exit
Choose an option: 1
Task title: python internship
Task description: internship

Task Manager
1. Add Task
2. Edit Task
3. Delete Task
4. Mark Task Complete
5. View Tasks
6. Exit
Choose an option: 1
Task title: exam preparation
Task description: prepare for python exam

Task Manager
1. Add Task
2. Edit Task
3. Delete Task
4. Mark Task Complete
5. View Tasks
6. Exit
Choose an option: 1
Task title: project submission
Task description: -
```

```
Task Manager
1. Add Task
2. Edit Task
3. Delete Task
4. Mark Task Complete
5. View Tasks
6. Exit
Choose an option: 5
1. python internship - internship [incomplete]
2. exam preparation - prepare for python exam [incomplete]
3. project submission - - [incomplete]

Task Manager
1. Add Task
2. Edit Task
3. Delete Task
4. Mark Task Complete
5. View Tasks
6. Exit
Choose an option: 4
Task number to mark complete: 2

Task Manager
1. Add Task
2. Edit Task
3. Delete Task
4. Mark Task Complete
5. View Tasks
6. Exit
Choose an option: 5
1. python internship - internship [incomplete]
2. exam preparation - prepare for python exam [COMPLETE]
3. project submission - - [incomplete]
```

```
Task Manager
1. Add Task
2. Edit Task
3. Delete Task
4. Mark Task Complete
5. View Tasks
6. Exit
Choose an option: 2
Task number to edit: 1
New title: motioncut python internship
New description: internship

Task Manager
1. Add Task
2. Edit Task
3. Delete Task
4. Mark Task Complete
5. View Tasks
6. Exit
Choose an option: 5
1. motioncut python internship - internship [incomplete]
2. exam preparation - prepare for python exam [COMPLETE]
3. project submission - - [incomplete]

Task Manager
1. Add Task
2. Edit Task
3. Delete Task
4. Mark Task Complete
5. View Tasks
6. Exit
Choose an option: 3
Task number to delete: 2
```

```
Task Manager
1. Add Task
2. Edit Task
3. Delete Task
4. Mark Task Complete
5. View Tasks
6. Exit
Choose an option: 5
1. motioncut python internship - internship [incomplete]
2. project submission - - [incomplete]

Task Manager
1. Add Task
2. Edit Task
3. Delete Task
4. Mark Task Complete
5. View Tasks
6. Exit
Choose an option: 6
None
```

HANGMAN GAME

Description :

1. In a hangman game, the player guesses a hidden word chosen by the computer.
2. The player can select a difficulty level. Easy, Medium or hard, each with different word lists.
3. The player guesses one letter at a time, and the correct guesses are revealed.
4. The player can make up to 6 incorrect guesses before losing the game.
5. The game provides feedback for each guess and ends when the word is guessed correctly or when the player loses.

```
import random
print("!!WELCOME TO HANGMAN GAME!!")
print("Do you want to go through rules??\nIf yes, press 1 to continue. Else press 2: \n")
rule = int(input())
if rule == 1:
    print("Here are the rules of the game:")
    print("1. The game is played between a computer and a user")
    print("2.Upon selecting the difficulty, computer chooses a word at random")
    print("3. The user has to guess the word by suggesting letters, one at a time")
    print("4. If the user guesses a letter correctly, it will be displayed in the correct
position")
    print("5. Maximum number of wrong guesses you can make is 6")
    print("ALL THE BEST!\n")

print("Choose Difficulty level(1-3): ")
print("1.Easy\n2.Medium\n3.Hard")
difficulty = int(input("Enter your choice: "))
if difficulty == 1:
    list1=["apple","banana","orange","cat","dog","chair"]
    comp_choice1 = random.choice(list1)
    word = comp_choice1
    len1 = len(comp_choice1)
    print("_"*len1)
    start = "_"*len1
    max_wrong = 6
    wrongguess = 0
    while wrongguess != max_wrong:
        letter1 = input("Guess a letter: ")
        if len(letter1) != 1:
            print("Please enter a single letter")
        else:
            if letter1 in comp_choice1:
                print("Good guess")
                list_start = list(start)
                for i in range(len(word)):
                    if word[i] == letter1:
                        list_start[i] = letter1
                start = ''.join(list_start)
                print(start)
                #print("\n")
            else:
                print("Bad guess")
                wrongguess += 1
                print(start)
                print("Wrong guesses remaining: %d\n"%(max_wrong-wrongguess))
            if start == comp_choice1:
                print("You found it!")
                exit(0)
    print("You lost!!. The word is %s"%(word))

elif difficulty == 2:
    list2=["france","india","japan","head","heart","salad"]
    comp_choice2 = random.choice(list2)
    word = comp_choice2
    len2 = len(comp_choice2)
    print("_"*len2)
    start = "_"*len2
    max_wrong = 6
    wrongguess = 0
    while wrongguess != max_wrong:
        letter2 = input("Guess a letter: ")
        if len(letter2) != 1:
            print("Please enter a single letter")
        else:
            if letter2 in comp_choice2:
                print("Good guess")
                list_start = list(start)
```

```
                for i in range(len(word)):
                    if word[i] == letter2:
                        list_start[i] = letter2
                start = ''.join(list_start)
                print(start)
                #print("\n")
            else:
                print("Bad guess")
                wrongguess += 1
                print(start)
                print("Wrong guesses remaining: %d\n"%(max_wrong-wrongguess))
            if start == comp_choice2:
                print("You found it!")
                exit(0)
    print("You lost!!. The word is %s"%(word))

elif difficulty == 3:
    list3=["freedom","biology","tennis","piano","basketball","pyramids"]
    comp_choice3 = random.choice(list3)
    word = comp_choice3
    len3 = len(comp_choice3)
    print("_"*len3)
    start = "_"*len3
    max_wrong = 6
    wrongguess = 0
    while wrongguess != max_wrong:
        letter3 = input("Guess a letter: ")
        if len(letter3) != 1:
            print("Please enter a single letter")
        else:
            if letter3 in comp_choice3:
                print("Good guess")
                list_start = list(start)
                for i in range(len(word)):
                    if word[i] == letter3:
                        list_start[i] = letter3
                start = ''.join(list_start)
                print(start)
                #print("\n")
            else:
                print("Bad guess")
                wrongguess += 1
                print("Wrong guesses remaining: %d\n"%(max_wrong-wrongguess))
            if start == comp_choice3:
                print("You found it!")
                exit(0)
    print("You lost!!. The word is %s"%(word))

else:
    print("Select difficulty level between 1 to 3!")
```

```
!!WELCOME TO HANGMAN GAME!!
Do you want to go through rules??
If yes, press 1 to continue. Else press 2:

1
Here are the rules of the game:
1. The game is played between a computer and a user
2.Upon selecting the difficulty, computer chooses a word at random
3. The user has to guess the word by suggesting letters, one at a time
4. If the user guesses a letter correctly, it will be displayed in the correct position
5. Maximum number of wrong guesses you can make is 6
ALL THE BEST!

Choose Difficulty level(1-3):
1.Easy
2.Medium
3.Hard
Enter your choice: 1

______
Guess a letter: b
Good guess
b_____
Guess a letter: c
Bad guess
b_____
Wrong guesses remaining: 5

Guess a letter: f
Bad guess
b_____
Wrong guesses remaining: 4
```

```
Guess a letter: n
Good guess
b_n_n_
Guess a letter: g
Bad guess
b_n_n_
Wrong guesses remaining: 3

Guess a letter: k
Bad guess
b_n_n_
Wrong guesses remaining: 2

Guess a letter: a
Good guess
banana
You found it!
```

```
!!WELCOME TO HANGMAN GAME!!
Do you want to go through rules??
If yes, press 1 to continue. Else press 2:

2
Choose Difficulty level(1-3):
1.Easy
2.Medium
3.Hard
Enter your choice: 2

____
Guess a letter: h
Good guess
h___
Guess a letter: t
Bad guess
h___
Wrong guesses remaining: 5

Guess a letter: a
Good guess
h_a_
Guess a letter: e
Good guess
hea_
Guess a letter: k
Bad guess
hea_
Wrong guesses remaining: 4

Guess a letter: h
Good guess
hea_
```

```
Guess a letter: n
Bad guess
hea_
Wrong guesses remaining: 3

Guess a letter: l
Bad guess
hea_
Wrong guesses remaining: 2

Guess a letter: t
Bad guess
hea_
Wrong guesses remaining: 1

Guess a letter: d
Good guess
head
You found it!
```

```
!!WELCOME TO HANGMAN GAME!!
Do you want to go through rules??
If yes, press 1 to continue. Else press 2:

2
Choose Difficulty level(1-3):
1.Easy
2.Medium
3.Hard
Enter your choice: 3

_______
Guess a letter: f
Bad guess
Wrong guesses remaining: 5

Guess a letter: b
Good guess
b______
Guess a letter: y
Good guess
b_____y
Guess a letter: d
Bad guess
Wrong guesses remaining: 4

Guess a letter: s
Bad guess
Wrong guesses remaining: 3

Guess a letter: k
Bad guess
Wrong guesses remaining: 2
```

```
Guess a letter: l
Good guess
b__l__y
Guess a letter: m
Bad guess
Wrong guesses remaining: 1

Guess a letter: o
Good guess
b_olo_y
Guess a letter: e
Bad guess
Wrong guesses remaining: 0

You lost!!. The word is biology
```

RANDOM PASSWORD GENERATOR

Description :

1. This program generates random passwords based on user input.
2. Users can choose to generate a password with all characters or digits based on specific criteria (digits, punctuation, etc.).
3. The length of password is defined by the user.
4. After generating the password, users are asked is they want to copy it to the clipboard using the pyperclip module.
5. The program handles invalid input and ensures the password is copied when requested.

```
import string
import random
import pyperclip

print("Choose an option:")
print("1.Generate a random password using all characters\n2.Generate a random password using
selecive criteria")

choiceinput = int(input())
if choiceinput == 1:
    try:
        length = int(input("Enter password length: "))
    except ValueError:
        print("Invalid input. Please enter an integer value.")
        exit()
    res = ''.join(random.choices(string.printable, k=length))
    print("Generated random password:", res)

elif choiceinput == 2:
    try:
        print("Select an option for password generation:")
        print("1: Digits only")
        print("2: Punctuation only")
        print("3: Uppercase letters only")
        print("4: Lowercase letters only")
        print("5: Digits and punctuation")
        print("6: Digits and letters")
        print("7: Uppercase and lowercase letters")
        selectinput = int(input("Enter your choice (1-7): "))

        print("Enter password length: ")
        pwdlen= int(input())
        if selectinput == 1:
            res = ''.join(random.choices(string.digits, k=pwdlen))
        elif selectinput == 2:
            res = ''.join(random.choices(string.punctuation, k=pwdlen))
        elif selectinput == 3:
            res = ''.join(random.choices(string.ascii_uppercase, k=pwdlen))
        elif selectinput == 4:
            res = ''.join(random.choices(string.ascii_lowercase, k=pwdlen))
        elif selectinput == 5:
            res = ''.join(random.choices(string.digits + string.punctuation, k=pwdlen))
        elif selectinput == 6:
            res = ''.join(random.choices(string.digits + string.ascii_letters, k=pwdlen))
        elif selectinput == 7:
            res = ''.join(random.choices(string.ascii_uppercase + string.ascii_lowercase,
k=pwdlen))
        else :
            print("Enter value from given combinations!")
        print("Generated random password is:",res)
    except ValueError:
        print("Invalid input. Please enter an integer value.")
        exit()

else:
    print("Choose a valid option!")


print("Would you like to copy the password?")
print("1.Yes\n2.No")
try:
    user_input = int(input())
    if user_input == 1:
        pyperclip.copy(res)
        copied_text = pyperclip.paste()
        print("Copied password:", copied_text)
    else:
```

```
        print("Password not copied.")
except ValueError:
    print("Invalid input. Please enter a valid number.")
```

```
Choose an option:
1.Generate a random password using all characters
2.Generate a random password using selecive criteria
1
Enter password length: 10
Generated random password: 2L="onU#8#
Would you like to copy the password?
1.Yes
2.No
1
Copied password: 2L="onU#8#
```

```
Choose an option:
1.Generate a random password using all characters
2.Generate a random password using selecive criteria
2
Select an option for password generation:
1: Digits only
2: Punctuation only
3: Uppercase letters only
4: Lowercase letters only
5: Digits and punctuation
6: Digits and letters
7: Uppercase and lowercase letters
Enter your choice (1-7): 1
Enter password length:
6
Generated random password is: 856682
Would you like to copy the password?
1.Yes
2.No
1
Copied password: 856682
PS C:\Users\chips\nihanth> 856682
```

```
Choose an option:
1.Generate a random password using all characters
2.Generate a random password using selecive criteria
2
Select an option for password generation:
1: Digits only
2: Punctuation only
3: Uppercase letters only
4: Lowercase letters only
5: Digits and punctuation
6: Digits and letters
7: Uppercase and lowercase letters
Enter your choice (1-7): 5
Enter password length:
9
Generated random password is: ,>9!735``
Would you like to copy the password?
1.Yes
2.No
2
Password not copied.
```

```
Choose an option:
1.Generate a random password using all characters
2.Generate a random password using selecive criteria
2
Select an option for password generation:
1: Digits only
2: Punctuation only
3: Uppercase letters only
4: Lowercase letters only
5: Digits and punctuation
6: Digits and letters
7: Uppercase and lowercase letters
Enter your choice (1-7): 7
Enter password length:
10
Generated random password is: ViAPuvXASD
Would you like to copy the password?
1.Yes
2.No
1
Copied password: ViAPuvXASD
```