Computer Science Practical File

Roll No: 13657779

Class: 12

Submitted by: Yuvraj Singh Cheema

Submitted to: Ms. Prabhjit Kaur

1. Program to find if the number entered is prime or not

Code:

```
def isPrime(number):
    if number == 1:
        return False
    for i in range (2,number):
        if number%i == 0:
            return False
    else:
        return True

num = int(input("Please enter a number: "))
if isPrime(num):
    print(num,"is a prime number")
else:
    print(num,"is not a prime number")
```

Output:

```
Please enter a number: 7

7 is a prime number

Please enter a number: 4

4 is not a prime number
```

2. Program to find factorial of a number

Code:

```
print('Welcome to factorial finder')
run = True
while run: # main loop
    good_inp = False
    while not good_inp: # loop to get an input with fail-safes for things a user can do
        try:
            num = int(input("Please enter a number:"))
            if num < 1: # Factorial doesn't exist for numbers below 1
                print("Error, bad input")
            else:
                good_inp = True
        except ValueError: # Fail-safe in case the user enters something other than a number
            print("I said number!!")

    fac = 1 # Default valur for factorial
    for i in range(1,num+1): # For loop to go over each number and multiply it to the factorial output
        fac *= i

    print("The factorial of "+str(num)+ " is:",fac)

    # code segment to prevent the program closing after just one input
    cont = input("Press enter to continue \nPress Q to quit ")
    try:
        if cont.upper()[0] == 'Q':
            run = False
    except IndexError:
        run = True
```

Output:

```
Welcome to factorial finder

Please enter a number:3

The factorial of 3 is: 6

Press enter to continue

Press Q to quit
```

```
Please enter a number:25

The factorial of 25 is: 15511210043330985984000000

Press enter to continue

Press Q to quit q
```

3. *Program to find the sum of elements of a list*

Code:

```
from random import randint # Importing randint from random to help make
the list
num_list = [randint(0,50) for i in range (10)] # Making the list using
list comprehension

sum = 0
for num in num_list:
    sum += num

print("The sum of all elements of",num_list,'is',sum)
```

Output:

```
The sum of all elements of [14, 41, 1, 24, 16, 46, 28, 27, 38, 14] is
249
```

4. Program to calculate the nth element of fibbonachi series

Code:

```
from random import randint # Importing randint from random to get the value of n
n = randint(0,50)
first = 0
second = 1

if n == 0:
    nthTerm = 0
elif n == 1:
    nthTerm = 1
else:
    for i in range(n - 2):
        first, second = second, first + second
    nthTerm = second

print(n, nthTerm)
```

Output:

```
15 377
```

5. Program to search a word in a given string

Code:

```
givenString = open('Text file').read()

toSearch = input('Word to search: ')

if toSearch in givenString:
    print(toSearch,'exists in the string at
index',givenString.index(toSearch))
else:
    print('The search phrase does not exist in the string')
```

Output:

```
Word to search: but

but exists in the string at index 38
```

6. Program to read content of a file and print it with spaces replaced with # symbol

Code:

```
file = open("Textfile")# opening the file
content = file.read() # saving the content of the file into a variable
words = content.split() # splitting the content by whitespaces using the builtin split() function

# Code to print the content
print(words[0], end='')
for i in words[1:]:
    print('#'+i, end='')
```

Output:

```
This#file#should#be#full#of#gibberish#but#I#am#writing#this#sentence#for
#it#to#look#professional
```

7. Program to read the content of file and display the total number of consonants, vovels and uppercase letters

Code:

```python
file = open('Text file') # opening the file
content = file.read() # saving the content of the file into a variable
vovels = ['A', 'E', 'I', 'O', 'U'] # list defining what are vovels
consonats = ['B', 'C', 'D', 'F', 'G', 'H', 'J', 'K', 'L', 'M', 'N', 'P',
'Q', 'R', 'S', 'T', 'V', 'W', 'X',' Y', 'Z'] # list defining what are
consonats

# Default values for the counting
num_vovels = 0
num_consonats = 0
num_upper = 0

# For loop for counting
for letter in content:
    if letter.isupper():
        num_upper += 1
    if letter.upper() in consonats:
        num_consonats += 1
    if letter.upper() in vovels:
        num_vovels += 1

# Printing result
print("Vovels =", num_vovels)
print("Consonats =", num_consonats)
print("Uppercase =", num_upper)
```

Output:

```
Vovels = 30

Consonats = 49

Uppercase = 2
```

8. To check whether a number is odd or even

Code:

```
def isEven(number):
    if number%2 == 0:
            return True
    else:
        return False

num = int(input("Please enter a number: "))
if isEven(num):
    print(num,"is an even number")
else:
    print(num,"is an odd number")
```

Output:

```
Please enter a number: 19

19 is an odd number
```

9. Program to find the occurrence of any word in a string

Code:

```
def counter(string,search):
    count = 0
    words = string.split()
    for word in words:
        if word == search:
            count += 1
    return count

sentence = input("Please enter a sentence: ")
word = input("Please enter what to search for in the sentence: ")

print(word, "was found", counter(sentence,word), "times.")
```

Output:

```
Please enter a sentence: The quick brown fox jumped over the lazy dog

Please enter what to search for in the sentence: cat

cat was found 0 times.
```

10. Binary file to save rollno and name

Code:

```
import pickle

def enter(file):
    f = open(file,'wb')
    try:
        student = pickle.load(open(file,'rb'))
    except EOFError:
        student = []

    try:
        rollNo =  input('Please enter the rollnumber: ')
        name = input("Please enter the name: ")
        student.append([rollNo,name])
        pickle.dump(student,f)
    except KeyboardInterrupt:
        print("cancelled")

def reader(file):
    f = open(file,'rb')
    student = pickle.load(f)
    try:
        r = input("Please enter roll number to search: ")
        for data in student:
            if data[0] == r:
                print("Name is",data[1])
                break
        else:
            print("Roll number not found")
    except KeyboardInterrupt:
        print("cancelled")

while True:
    usr = input("Press r to read and w to write information: ")
    if usr.lower() == "r":
        print("Press ctrl + c to cancel")
        reader("student.dat")
    elif usr.lower() == "w":
        print("Press ctrl + c to cancel")
        enter("student.dat")
    else:
        print("Error: Wrong input")
```

Output:

```
Press r to read and w to write information: w

Press ctrl + c to cancel

Please enter the rollnumber: 1

Please enter the name: Jasdish

Press r to read and w to write information: r

Press ctrl + c to cancel

Please enter roll number to search: 1

Name is Jasdish

Press r to read and w to write information: w

Press ctrl + c to cancel

Please enter the rollnumber: 2

Please enter the name: Prateekpal

Press r to read and w to write information: r

Press ctrl + c to cancel

Please enter roll number to search: 2

Name is Prateekpal
```

11. Read content of a file and copy the lines that do not contain 'a' to another file

Code:

```
f1 = open("See you again.txt","r")
f2 = open("File2","w")

for line in f1:
    if "a" not in line:
        f2.write(line)

f1.close()
f2.close()
```

Output:

```
Uh

How can we not talk about family when family's all that we
got?

Everything I went through, you were standing there by my side

And now you gon' be with me for the last ride

It's been a long day without you, my friend

And I'll tell you all about it when I see you again (I'll see you
again)

We've come a long way (yeah, we came a long way)

From where we began (you know we started)

Oh, I'll tell you all about it when I see you again (I'll tell you)

When I see you again

First, you both go out your way and the vibe is feeling strong

And what's small turned to a friendship, a friendship turned to
a bond

And that bond will never be broken, the love will never get lost

(The love will never get lost)

And when brotherhood come first, then the line will never be
crossed

Established it on our own when that line had to be drawn

And that line is what we reached, so remember me when I'm
gone

(Remember me when I'm gone)
```

```
Uh

(The love will never get lost)

And when brotherhood come first, then
the line will never be crossed

(Remember me when I'm gone)
```

12. Dice Simulation

Code:

```
import random

while True:
    try:
        input("Press enter to generate a number")
        print(random.randint(1,6))
    except KeyboardInterrupt:
        print('Goodbye')
        break
```

Output:

```
Press enter to generate a number

1

Press enter to generate a number

5

Press enter to generate a number

3

Press enter to generate a number

4

Press enter to generate a number

4

Press enter to generate a number

5

Press enter to generate a numberGoodbye
```

13. Implement stack using a list

Code:

```
def push(S,item):
    global top
    S.append(item)
    top = len(S) - 1

def pull(S):
    global top
    if len(S) != 0:
        val = S.pop()
    else:
        return None
    if len(S) == 0:
        top = None
    else:
        top = len(S)-1
    return val

def peek(S):
    if len(S) == 0:
        return None
    return S[top]

def show(S,top):
    if len(S) == 0:
        print("Stack is empty")
    else:
        print(" (TOP) <== ",end = "")
        while top >= 0:
            print(S[top],"<== ",end="")
            top -= 1
        print()

stack = []
top = None
while True:
    print("::Stack Demonstration::")
    print("1. PUSH")
    print("2. PULL")
    print("3. PEEK")
    print("4. SHOW")
    print("0. EXIT")
    try:
        usr = int(input())
        if usr == 1:
            item = input("Enter item to push: ")
            push(stack,item)
```

```python
            print(item,"successfully pushed")
        elif usr == 2:
            val = pull(stack)
            if val == None:
                print("Stack is empty")
            else:
                print(val,"removed from stack")
        elif usr == 3:
            val = peek(stack)
            if val == None:
                print("Stack is empty")
            else:
                print("The topmost value in the stack is",val)
        elif usr == 4:
            show(stack,top)
        elif usr == 0:
            break
        else:
            print("Invalid input")
    except ValueError:
        print("Invalid input")
```

Output:

```
::Stack Demonstration::

1. PUSH

2. PULL

3. PEEK

4. SHOW

0. EXIT

1

Enter item to push: 25

25 successfully pushed

::Stack Demonstration::

1. PUSH

2. PULL

3. PEEK

4. SHOW

0. EXIT

1
```

```
Enter item to push: 86
86 successfully pushed
::Stack Demonstration::
1. PUSH
2. PULL
3. PEEK
4. SHOW
0. EXIT
4
(TOP) <== 86 <== 25 <==
::Stack Demonstration::
1. PUSH
2. PULL
3. PEEK
4. SHOW
0. EXIT
3
The topmost value in the stack is 86
::Stack Demonstration::
1. PUSH
2. PULL
3. PEEK
4. SHOW
0. EXIT
2
86 removed from stack
::Stack Demonstration::
1. PUSH
2. PULL
3. PEEK
4. SHOW
0. EXIT
```

```
25
Invalid input
::Stack Demonstration::
1. PUSH
2. PULL
3. PEEK
4. SHOW
0. EXIT
4
(TOP) <== 25 <==
::Stack Demonstration::
1. PUSH
2. PULL
3. PEEK
4. SHOW
0. EXIT
0
```

14. Implement queue using list

Code:

```
def isEmpty(q):
    return not bool(len(q))

def add(q,item):
    q.append(item)
    print(item, "successfully added.")

def remove(q):
    if not isEmpty(q):
        val = q.pop(0)
        return val

def peek(q):
    if not isEmpty(q):
        return q[0]

def show(q):
    if not isEmpty(q):
        print("Front <= ",end="")
        for i in q:
            print(i,"<= ",end="")
        print("Rear")
        return 0

Q=[]
while True:
    print("::Queue::")
    print("1. Add to Queue")
    print("2. Progress Queue")
    print("3. Show Next")
    print("4. Show Queue")
    print("0. Exit")
    try:
        usr = int(input())
        if usr == 1:
            item = input("Enter item to add: ")
            add(Q, item)
        elif usr == 2:
            val = remove(Q)
            if val == None:
                print("Queue is empty")
            else:
                print(val,"removed successfully")
        elif usr == 3:
            next = peek(Q)
            if next != None:
                print("Next in queue:",next)
            else:
                print("Queue is empty")
        elif usr == 4:
            temp = show(Q)
            if temp == None:
                print("Queue is empty")
        elif usr == 0:
            break
        else:
            print("Invalid input")
    except ValueError:
        print("Invalid input")
```

Output:

```
::Queue::
1. Add to Queue
2. Progress Queue
3. Show Next
4. Show Queue
0. Exit
1
Enter item to add: 3
3 successfully added.
::Queue::
1. Add to Queue
2. Progress Queue
3. Show Next
4. Show Queue
0. Exit
1
Enter item to add: 7
7 successfully added.
::Queue::
1. Add to Queue
2. Progress Queue
```

```
3. Show Next
4. Show Queue
0. Exit
2
3 removed successfully
::Queue::
1. Add to Queue
2. Progress Queue
3. Show Next
4. Show Queue
0. Exit
3
Next in queue: 7
::Queue::
1. Add to Queue
2. Progress Queue
3. Show Next
4. Show Queue
0. Exit
1
Enter item to add: 4
4 successfully added.
::Queue::
1. Add to Queue
2. Progress Queue
3. Show Next
4. Show Queue
0. Exit
3
Next in queue: 7
::Queue::
1. Add to Queue
```

```
2. Progress Queue

3. Show Next

4. Show Queue

0. Exit

4

Front <= 7 <= 4 <= Rear

::Queue::

1. Add to Queue

2. Progress Queue

3. Show Next

4. Show Queue

0. Exit

0
```

15. Take sample phishing emails and find most common hostname

Code:

```
email = ["jackpotwin@lottery.com",
         "claimtheprize@mymoney.com",
         "youarethewinner@lottery.com",
         "luckywinner@mymoney.com",
         "spinthewheel@flipkart.com",
         "dealwinner@snapdeal.com",
         "luckywinner@snapdeal.com",
         "luckyjackpot@americanlottery.com",
         "claimtheprize@lootolottery.com",
         "youarelucky@mymoney.com"]

hosts = {}

for i in email:
    host = i.split('@')[1]
    if host not in hosts:
        hosts[host] = 1
    else:
        hosts[host] += 1

print(max(hosts,key=hosts.get))
```

Output:

```
mymoney.com
```

16. Stick picking game such that computer wins everytime

Code:

```python
def draw(n):
    print("o  "*n)
    print("|  "*n)
    print("|  "*n)
    print("|  "*n)
    print("|  "*n)

total = 21
pick = 0
player = True
win = False

print("Welcome To Stick Picking Game : Computer Vs User")
print("Rule: 1) Both User and Computer can pick sticks between 1 to 4 at a time")
print("      2) Whosoever picks the last stick will be the looser")
print("++ Lets Begin ++")


draw(total)
while not win:
    if player:
        pick = 0
        print("++ User Turn ++")
        while pick not in [1,2,3,4]:
            pick = int(input("User pick the number of sticks you want to remove: "))
        total -= pick
        draw(total)
        player = False
    else:
        print("++ Computer Turn ++")
        comp = 5-pick
        print("Computer removed",comp,"sticks")
        total -= comp
        draw(total)
        if total == 1:
            win = True
            print("Winner -- Computer")
            print("Better luck next time'")
        player = True
```

Output:

```
Welcome To Stick Picking Game : Computer Vs User
Rule: 1) Both User and Computer can pick sticks between 1 to 4 at a time
      2) Whosoever picks the last stick will be the looser
++ Lets Begin ++
o o o o o o o o o o o o o o o o o o o o o
| | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | |
++ User Turn ++
User pick the number of sticks you want to remove: 1
o o o o o o o o o o o o o o o o o o o o
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
++ Computer Turn ++
Computer removed 4 sticks
o o o o o o o o o o o o o o o o
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
++ User Turn ++
User pick the number of sticks you want to remove: 3
```

```
o o o o o o o o o o o o o
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
++ Computer Turn ++
Computer removed 2 sticks
o o o o o o o o o o o
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
++ User Turn ++
User pick the number of sticks you want to remove: 4
o o o o o o o
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
++ Computer Turn ++
Computer removed 1 sticks
o o o o o o
| | | | | |
| | | | | |
| | | | | |
| | | | | |
++ User Turn ++
User pick the number of sticks you want to remove: 2
o o o o
| | | |
| | | |
| | | |
| | | |

++ Computer Turn ++

Computer removed 3 sticks

o

|

|

|

|

Winner -- Computer

Better luck next time
```

17. Using OOP to simulate bank account

Code:

```python
class BankAccount:
  def __init__(self):
    self.balance = 0

  def deposit(self, amount):
    self.balance += amount
    return self.balance

  def withdraw(self, amount):
    if self.balance - amount < 0:
      return "Insufficient Funds"
    self.balance -= amount
    return self.balance

myAccount = BankAccount()

myAccount.deposit(2000)
print("Balance =", myAccount.balance)
myAccount.withdraw(500)
print("Balance =", myAccount.balance)
```

Output:

```
Balance = 2000

Balance = 1500
```

18. Check if given string is a palindrome

Code:

```python
def pali(word):
    reverse = word[::-1]
    return word.lower() == reverse.lower()

print("bye :",pali("bye"))
print("aibohphobia :",pali("aibohphobia"))
print("plaidrome :",pali("plaidrome"))
print("racecar :",pali("racecar"))
```

Output:

```
bye : False

aibohphobia : True

plaidrome : False

racecar : True
```

19. Find highest common factor using recursion

Code:

```
def hcf(a,b):
    if (b==0):
        return a
    else:
        return hcf(b, a%b)

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))

print("HCF of",num1,"&",num2,"is",hcf(num1,num2))
```

Output:

```
Enter first number: 17

Enter second number: 51

HCF of 17 & 51 is 17
```

20. Creating lists using list comprehension

Code:

```
num = [i for i in range(0,20)]
mul = [2**i for i in range(0,20)]

for i in range(0,20):
    print(num[i]," : ",mul[i])
```

Output:

```
0  :  1
1  :  2
2  :  4
3  :  8
4  :  16
5  :  32
6  :  64
7  :  128
8  :  256
9  :  512
10  :  1024
11  :  2048
12  :  4096
13  :  8192
14  :  16384
15  :  32768
16  :  65536
17  :  131072
18  :  262144
19  :  524288
```