

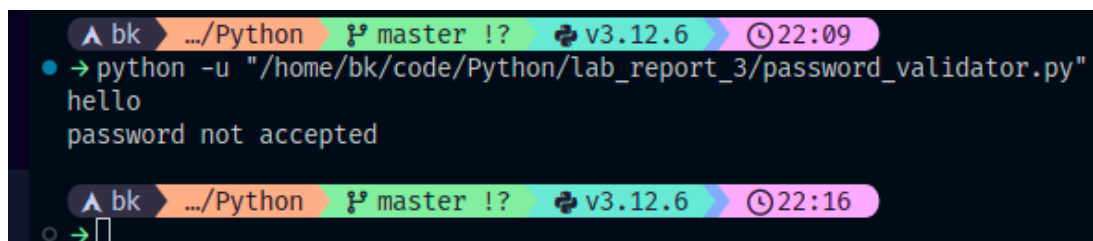
## 1. Password validator

```
length,upper,lower,num=0,0,0,0

password = input()

for i in range(len(password)):
    if len(password)>=8:
        length=1
    if password[i].isupper():
        upper=1
    if password[i].islower():
        lower=1
    if password[i].isnumeric():
        num=1

if (length+upper+lower+num)>=4:
    print("password accepted")
else:
    print("password not accepted")
```



The screenshot shows a terminal window with a dark background. The prompt is 'bk' and the directory is '.../Python'. The user has entered 'python -u "/home/bk/code/Python/lab\_report\_3/password\_validator.py"', and the output is 'hello' followed by 'password not accepted'. The terminal shows the script was run at 22:09 and 22:16.

```
bk > .../Python master !? v3.12.6 22:09
→ python -u "/home/bk/code/Python/lab_report_3/password_validator.py"
hello
password not accepted

bk > .../Python master !? v3.12.6 22:16
→
```

## 2. Divisible by 3 or 5

```
for i in range(1,51):
```

```

    if i%3 != 0 and i%5 !=0:
        continue
    else:
        print(i,end=' ')

```

```

^ bk > .../Python master !? v3.12.6 22:16
• → python -u "/home/bk/code/Python/lab_report_3/divisible_by_3_5.py"
  3 5 6 9 10 12 15 18 20 21 24 25 27 30 33 35 36 39 40 42 45 48 50
^ bk > .../Python master !? v3.12.6 22:17
○ →

```

### 3. Positive number sum

```

sum=0
while True:
    num=float(input())
    if(num<0):
        break
    sum=sum+num

print(sum)

```

```

^ bk > .../Python master !? v3.12.6 22:18
• → python -u "/home/bk/code/Python/lab_report_3/possitive_number_sum.py"
  5
  6
  78
  3
  -1
  92.0
^ bk > .../Python master !? v3.12.6 22:18
○ →

```

### 4. Word palindrome

```

while True:

    str=input()

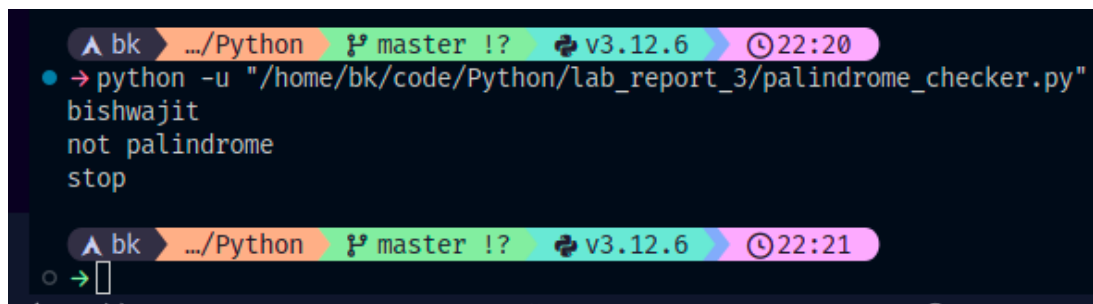
    if str=='stop':
        break

    if len(str)<3:
        print("skipped")
        continue

    for i in range(len(str)):
        if str[i]!=str[len(str)-1-i]:
            print("not palindrome")
            break

    print("palindrome")

```



A terminal window showing the execution of a Python script. The prompt is 'bk' and the directory is '.../Python'. The script is run with the command 'python -u "/home/bk/code/Python/lab\_report\_3/palindrome\_checker.py"'. The input 'bishwajit' results in the output 'not palindrome'. The input 'stop' results in the output 'stop'. The terminal shows the script's output for two different inputs.

```

bk > .../Python master !? v3.12.6 22:20
→ python -u "/home/bk/code/Python/lab_report_3/palindrome_checker.py"
bishwajit
not palindrome
stop

```

## 5. Vowel counter

```

str=input()

str=str.lower()

vowel_list=['a','e','i','o','u']

vowel_count=0

for i in range(len(str)):
    if str[i] not in vowel_list:

```

```

        continue

    else:

        vowel_count+=1

print(vowel_count)

```

```

^ bk  .../Python  master !?  v3.12.6  22:21
→ python -u "/home/bk/code/Python/lab_report_3/vowel_counter.py"
bishwajit
3

^ bk  .../Python  master !?  v3.12.6  22:22
→

```

## 6. Unique character

```

str=input()
str_set=set()
break_=False
index=0
for i in range(len(str)):
    if str[i] not in str_set:
        str_set.add(str[i])
    else:
        index=i
        break_=True
        break

if break_:
    print("repeat found at index",index)
else:
    print('repeat not found')

```

```
^ bk > .../Python  master !?  v3.12.6  22:22
• → python -u "/home/bk/code/Python/lab_report_3/unique_charecter.py"
bishwajit
repeat found at index 7

^ bk > .../Python  master !?  v3.12.6  22:23
○ →
```

## 7. Student information system

```
students = {}

def display_student_data():
    if len(students)==0:
        print("No Students")
        return

    for student in students.values():
        print(f"Name: {student['name']}")
        print(f"Roll No: {student['roll_no']}")
        print(f"Age: {student['age']:.2f}")

while True:
    print("\nChoose an option:")
    print("1. Add Student")
    print("2. Display Student Data")
    print("3. Exit")

    choice = input("Enter your choice (1/2/3): ")
    name_input=""
```

```

roll_no_input=0
age_input=0.00

if choice == '1':
    checker = int(input("enter 1 for enter name: "))
    if checker==1:
        name_input = input("Enter student's name: (max 5
characters): ")
    else:
        print("Invalid choice")
    checker = int(input("enter 1 for enter roll: "))
    if checker==1:
        roll_no_input = int(input("Enter student's Roll No: "))
    else:
        print("Invalid choice")
    checker = int(input("enter 1 for enter age: "))
    if checker==1:
        age_input = float(input("Enter student's age (float):
"))
    else:
        print("Invalid choice")

    students[roll_no_input] = { "name": name_input, "roll_no":
roll_no_input, "age": age_input }

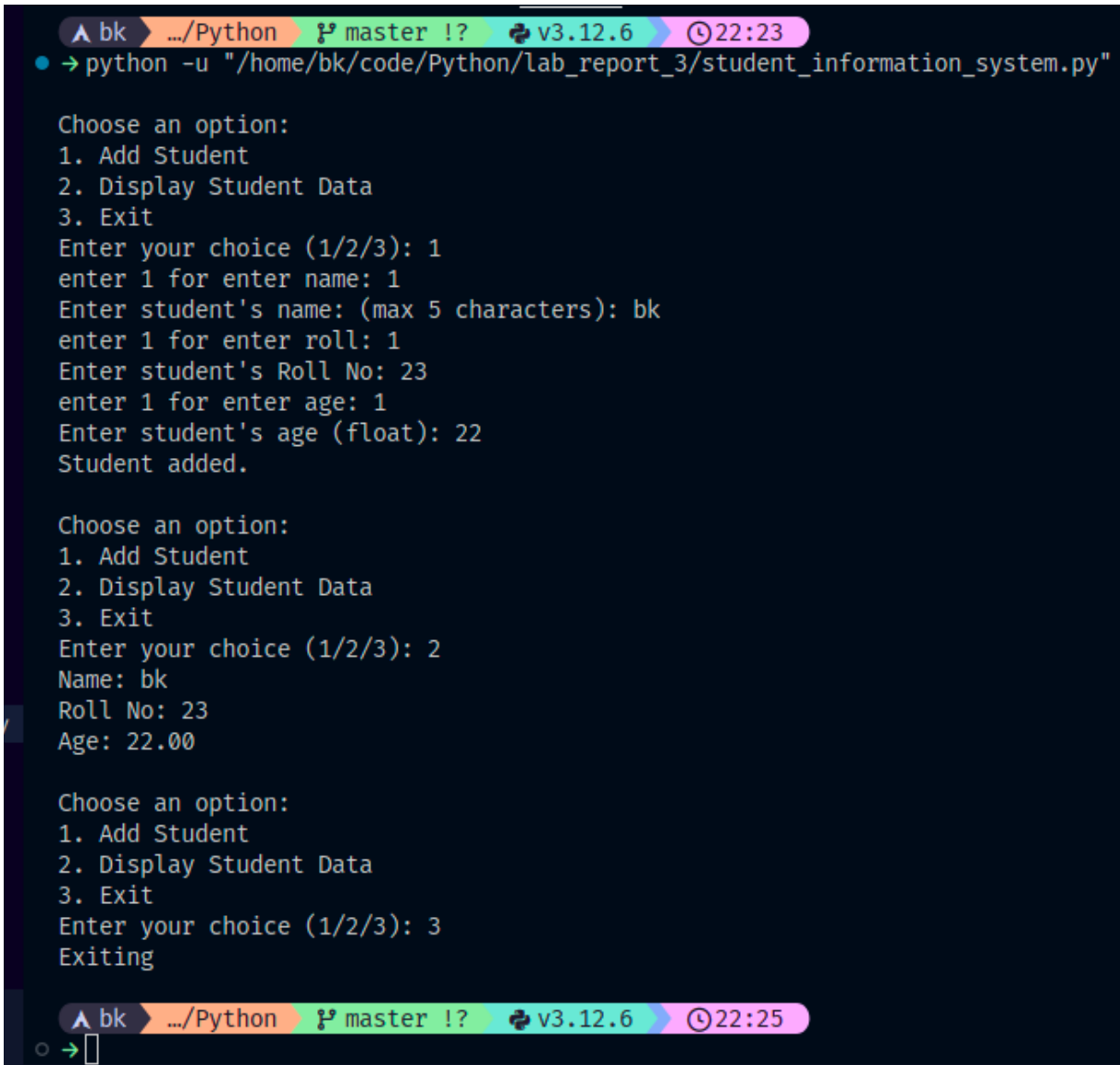
    print("Student added.")
elif choice == '2':
    display_student_data()
elif choice == '3':
    print("Exiting")

```

```
        break

    else:

        print("Invalid choice")
```



A terminal window with a dark background and a colorful status bar at the top. The status bar shows the user 'bk', the directory '.../Python', the shell 'master !?', the Python version 'v3.12.6', and the time '22:23'. The terminal shows the command to run a Python script: `python -u "/home/bk/code/Python/lab_report_3/student_information_system.py"`. The script prompts the user to 'Choose an option:' with a list: '1. Add Student', '2. Display Student Data', and '3. Exit'. The user enters '1'. The script then prompts for 'enter 1 for enter name: 1', 'Enter student's name: (max 5 characters): bk', 'enter 1 for enter roll: 1', 'Enter student's Roll No: 23', 'enter 1 for enter age: 1', and 'Enter student's age (float): 22'. It then prints 'Student added.'. The menu is shown again, and the user enters '2'. The script displays 'Name: bk', 'Roll No: 23', and 'Age: 22.00'. The menu is shown a third time, and the user enters '3'. The script prints 'Exiting'. The status bar at the bottom shows the time has changed to '22:25'.

```
bk .../Python master !? v3.12.6 22:23
→ python -u "/home/bk/code/Python/lab_report_3/student_information_system.py"

Choose an option:
1. Add Student
2. Display Student Data
3. Exit
Enter your choice (1/2/3): 1
enter 1 for enter name: 1
Enter student's name: (max 5 characters): bk
enter 1 for enter roll: 1
Enter student's Roll No: 23
enter 1 for enter age: 1
Enter student's age (float): 22
Student added.

Choose an option:
1. Add Student
2. Display Student Data
3. Exit
Enter your choice (1/2/3): 2
Name: bk
Roll No: 23
Age: 22.00

Choose an option:
1. Add Student
2. Display Student Data
3. Exit
Enter your choice (1/2/3): 3
Exiting

bk .../Python master !? v3.12.6 22:25
→
```

## 8. Grocery shop

```
grocery_items = {}
```

```
while True:
```

```
print("\nChoose an option:")
print("1. Add Item")
print("2. Remove Item")
print("3. View Basket")
print("4. Exit")

choice = input("Enter your choice (1/2/3/4): ")

if choice == '1':
    item = input("Enter item: ")
    if item in grocery_items:
        print("already exists")
    else:
        grocery_items[item] = 1
    print("item added")
elif choice == '2':
    item = input("Enter the item name to remove: ")
    if item not in grocery_items:
        print("item not found")
    else:
        del grocery_items[item]
    print("item removed")

elif choice == '3':
    if len(grocery_items)==0:
        print("empty")
    else:
```



```

        print("items list:")

        for item, count in grocery_items.items():

            print(f" {item}: {count}")

    elif choice == '4':

        print("Exiting")

        break

    else:

        print("invalid choice")

```

```

^ bk  .../Python  master !?  v3.12.6  22:25
→ python -u "/home/bk/code/Python/lab_report_3/grocery_shop.py"

Choose an option:
1. Add Item
2. Remove Item
3. View Basket
4. Exit
Enter your choice (1/2/3/4): 1
Enter item: 1
item added

Choose an option:
1. Add Item
2. Remove Item
3. View Basket
4. Exit
Enter your choice (1/2/3/4): 1
Enter item: 2
item added

Choose an option:
1. Add Item
2. Remove Item
3. View Basket
4. Exit
Enter your choice (1/2/3/4): 3
items list:
 1: 1
 2: 1

Choose an option:
1. Add Item
2. Remove Item
3. View Basket
4. Exit
Enter your choice (1/2/3/4): 4
Exiting

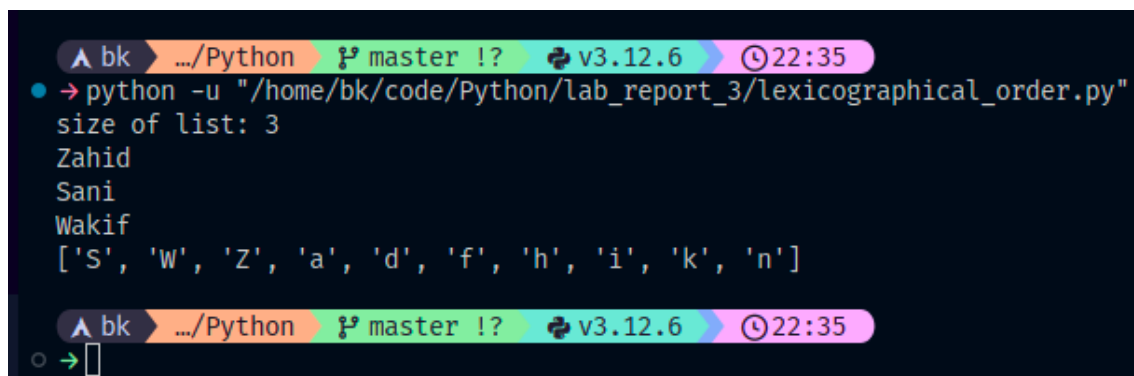
^ bk  .../Python  master !?  v3.12.6  22:30
→

```

## 9. Lexicographical order

```
list_size=int(input("size of list: "))
set_of_char=set()
for i in range(list_size):
    new_string=input()
    for char in new_string:
        if char not in set_of_char:
            set_of_char.add(char)

print(sorted(set_of_char))
```



```
bk ~/Python master !? v3.12.6 22:35
→ python -u "/home/bk/code/Python/lab_report_3/lexicographical_order.py"
size of list: 3
Zahid
Sani
Wakif
['S', 'W', 'Z', 'a', 'd', 'f', 'h', 'i', 'k', 'n']

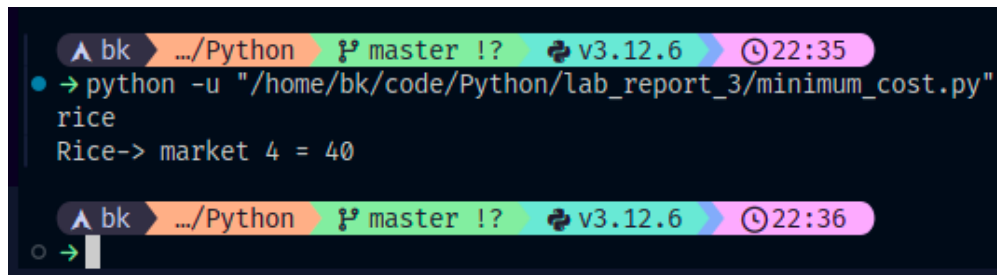
bk ~/Python master !? v3.12.6 22:35
→
```

## 10. Minimum cost

```
products = {
    "Rice": [45, 42, 41, 40],
    "Salt": [34, 35, 36, 36],
    "Fish": [200, 202, 201, 205],
    "Orange": [100, 99, 101, 102]
}

product=input().capitalize()
```

```
min_price=min(products[product])  
place_of_min=products[product].index(min_price)+1  
print(f"{product}-> market {place_of_min} = {min_price}")
```

A terminal window with a dark background. The top bar shows the user 'bk', the directory '.../Python', the shell 'master !?', the Python version 'v3.12.6', and the time '22:35'. The command 'python -u "/home/bk/code/Python/lab\_report\_3/minimum\_cost.py"' is entered and executed. The output shows 'rice' and 'Rice-> market 4 = 40'. The bottom bar shows the time updated to '22:36'.

```
bk .../Python master !? v3.12.6 22:35  
→ python -u "/home/bk/code/Python/lab_report_3/minimum_cost.py"  
rice  
Rice-> market 4 = 40  
bk .../Python master !? v3.12.6 22:36  
→
```

## 11. Email generator

```
student_number=int(input())  
  
for i in range(student_number):  
    name=input()  
    name=name.strip()  
    if(len(name)>20):  
        print("name has more than 20 char")  
        continue  
    lower=name.lower()  
    length=len(name)  
    first_char_ascii=ord(name[0])  
  
    email=f'{lower}_{length}_{first_char_ascii}'  
    print(email)
```

```
^ bk > .../Python master !? v3.12.6 22:37
• → python -u "/home/bk/code/Python/lab_report_3/email_generator.py"
3
bishwajit
bishwajit_9_98
joy
joy_3_106
hello
hello_5_104

^ bk > .../Python master !? v3.12.6 22:38
○ →
```

## 12. compressed string

```
number_of_string=int(input("enter how many string: "))
size_of_string=int(input("enter size of string: "))

compressed=[]

for i in range(number_of_string):
    string=input()
    if len(string)>size_of_string:
        string=string[:size_of_string]

    new_compressed=''
    count=1
    for i in range(1,len(string)):
        if string[i]==string[i-1]:
            count+=1
        else:
            if count >= 1:
                new_compressed+= str(count)+string[i-1]
            else:
```

```

        new_compressed+=string[i-1]

        count=1

    if count>=1:
        new_compressed+=str(count)+string[-1]
    else:
        new_compressed+=string[-1]

    compressed.append(new_compressed)

print(compressed)

```

```

A bk .../Python master !? v3.12.6 22:38
• → python -u "/home/bk/code/Python/lab_report_3/string_compressor.py"
enter how many string: 3
enter size of string: 6
ttyyuu
uuiokjj
rrss
['2t2y2u', '2u1i1o1k1j', '2r2s']

A bk .../Python master !? v3.12.6 22:40
○ → 

```

### 13. To do list

```

dict={}

while True:
    choice = int(input("1 for add task\n2 for mark task
completed\n3 for display all task\n--> "))
    i=1
    if choice==1:
        task=input("add your task: ")

```

```

        dict[i]=[task,False]

        i+=1

    elif choice==2:

        task_no=int(input("enter task no: "))

        dict_list=dict[task_no][1]=True

        print("update done")

        print(dict)

        print()

    elif choice==3:

        print()

        print(dict)

        print()

    else:

        print("wrong choice try again")

```

```

^ bk  .../Python  master !?  v3.12.6  22:40
○ → python -u "/home/bk/code/Python/lab_report_3/to_do_list.py"
1 for add task
2 for mark task completed
3 for display all task
--> 1
add your task: hello
1 for add task
2 for mark task completed
3 for display all task
--> 2
enter task no: 1
update done
{1: ['hello', True]}

1 for add task
2 for mark task completed
3 for display all task
--> 3

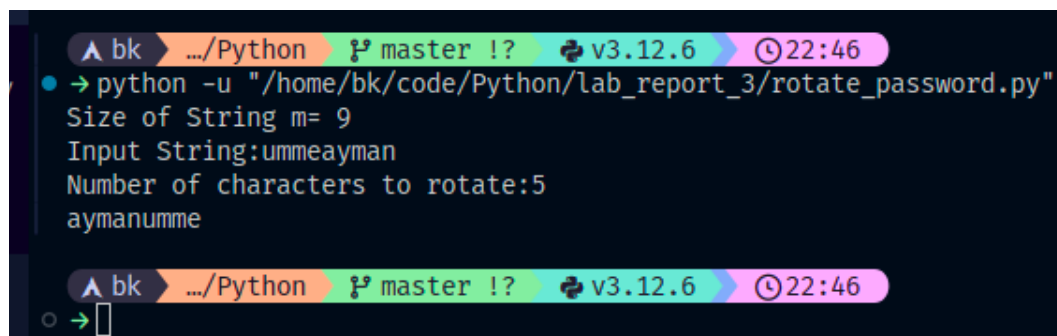
{1: ['hello', True]}

1 for add task
2 for mark task completed
3 for display all task
--> 

```

#### 14. Rotate password

```
m=int(input("Size of String m= "))
str=input("Input String:")
n=int(input("Number of characters to rotate:"))
final=""
if " " in str:
    print("no whitespace plz")
else:
    first_part=str[:n-1]
    second_part=str[n-1:]
    final=second_part+first_part
print(final)
```



The screenshot shows a terminal window with a dark background. The prompt is 'bk' and the directory is '.../Python'. The user has run the command 'python -u "/home/bk/code/Python/lab\_report\_3/rotate\_password.py"'. The program's output is as follows:

```
Size of String m= 9
Input String:ummeayman
Number of characters to rotate:5
aymanumme
```

Below the output, there is a new prompt 'bk' and a cursor, indicating the program has finished execution.

#### 15. Employee system

```
name_list=["Umme","Eity","Esrat"]

dict={}

asc_list=[]

for i in range(len(name_list)):
    sum=0
```

```

    for j in range(len(name_list[i])):
        asc_value=ord(name_list[i][j])

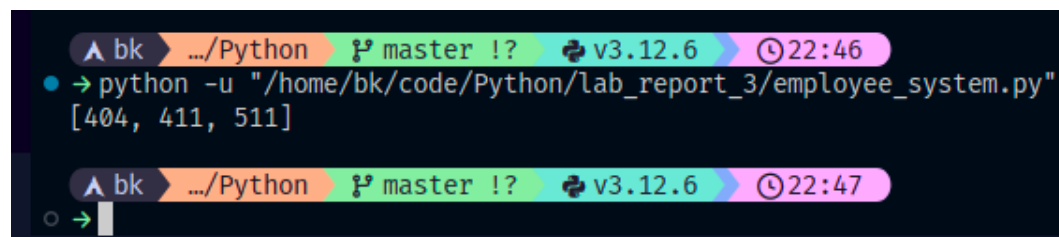
        sum=sum+asc_value

    dict[name_list[i]]=sum

    asc_list.append(sum)

print(asc_list)

```



```

^ bk > .../Python master !? v3.12.6 22:46
• → python -u "/home/bk/code/Python/lab_report_3/employee_system.py"
[404, 411, 511]

^ bk > .../Python master !? v3.12.6 22:47
○ →

```

## 16. Password maker

```

def pass_maker(str):

    if len(str)!=10:

        print("expected password of 10 length")

        return

    upper_str=str[:3].upper()

    lower_str=str[-3:].lower()

    special_str="!@#$"

    return upper_str+special_str+lower_str

str=input()

new_pass = pass_maker(str)

print(new_pass)

```



```
^ bk > .../Python master !? v3.12.6 22:50
• → python -u "/home/bk/code/Python/lab_report_3/password_maker.py"
bishwajitK
BIS!@#$itk

^ bk > .../Python master !? v3.12.6 22:50
○ →
```

## 17. Sum of even

```
x=[3,4,5,7,8,10,'dghf','cbxgc',3.5, 6.2]
sum=0
for i in range(len(x)):
    if type(x[i]) == int and x[i]%2==0:
        sum=sum+x[i]

print(sum)
```

```
^ bk > .../Python master !? v3.12.6 22:50
• → python -u "/home/bk/code/Python/lab_report_3/sum_of_even.py"
22

^ bk > .../Python master !? v3.12.6 22:51
○ →
```