<img src='https://www.darshan.ac.in/Content/media/DU_Logo.svg' width="250" height="300"/>

```
<center><h1>Python Programming - 2301CS404</center>
<center><h1>Lab - 3</center>
<center><h1>Charmi Bhalodiya</center>
<center><h1>23010101020</h1></center>
<center><h1>4B-448 8th batch</center>
```

In [1]:
```
# String
```

01) WAP to check whether the given string is palindrome or not.

In [15]:
```
str="HAMAH"
if(str[::-1]==str):
    print("string is palindrome")
else:
    print("string is not palindrome")
```

```
string is palindrome
```

02) WAP to reverse the words in the given string.

In [31]:
```
def reverse_words(string):
    words = string.split()
    reversed_words = " ".join(words[::-1])
    return reversed_words

string = input("Enter a string: ")
print("Reversed words:", reverse_words(string))
```

```
Reversed words: mahek is name my
```

03) WAP to remove ith character from given string.

In [29]:
```
def remove_ith_char(string, i):
    if 0 <= i < len(string):
        return string[:i] + string[i+1:]
    return string

string = input("Enter a string: ")
i = int(input("Enter the index to remove (0-based): "))
print("String after removing character:", remove_ith_char(string,i))
```

```
String after removing character: maek
```

04) WAP to find length of string without using len function.

In [33]:
```
def string_length(string):
    count = 0
    for _ in string:
        count += 1
    return count

string = input("Enter a string: ")
print("Length of the string:", string_length(string))
```

```
Length of the string: 8
```

05) WAP to print even length word in string.

In [39]:
```
def even_length_words(string):
    words = string.split()
    return [word for word in words if len(word) % 2 == 0]

string = input("Enter a string: ")
print("Even length words:", even_length_words(string))
```

```
Even length words: ['Mahekk', 'Gajjar']
```

06) WAP to count numbers of vowels in given string.

In [41]:
```
def count_vowels(string):
    vowels = "aeiouAEIOU"
    return sum(1 for char in string if char in vowels)

string = input("Enter a string: ")
print("Number of vowels:", count_vowels(string))
```

```
Number of vowels: 2
```

07) WAP to capitalize the first and last character of each word in a string.

In [45]:
```
def capitalize_first_last(string):
    words = string.split()
    modified_words = [
        word[0].upper() + word[1:-1] + word[-1].upper() if len(word) > 1 else word.upper()
        for word in words
    ]
    return " ".join(modified_words)

string = input("Enter a string: ")
print("Modified string:", capitalize_first_last(string))
```

```
Modified string: MY NamE IS MaheK
```

08) WAP to convert given array to string.

In [73]:
```
def array_to_string(arr):
    return " ".join(map(str, arr))

arr1 = ["Mahekk", "Are", "you", "Fine?"]
arr2 = ["No","I'm", "not"]
arr3 = ["Why?"]
arr4 = ["Bcz", "of", "my" ,"bestfrd"]
print("Vanshita:", array_to_string(arr1))
print("Nannii:", array_to_string(arr2))
print("Vanshita:", array_to_string(arr3))
print("Nannii:", array_to_string(arr4))
```

```
Vanshita: Mahekk Are you Fine?
Nannii: No I'm not
Vanshita: Why?
Nannii: Bcz of my bestfrd
```

09) Check if the password and confirm password is same or not.

In case of only case's mistake, show the error message.

In [77]:
```
def check_password(password, confirm_password):
    if password == confirm_password:
        return "Passwords match!"
    elif password.lower() == confirm_password.lower():
        return "Passwords do not match. Case mismatch detected!"
    else:
        return "Passwords do not match."

password = input("Enter password: ")
confirm_password = input("Enter confirm password: ")
print(check_password(password, confirm_password))
```

```
Passwords match!
```

10) : Display credit card number.

card no. : 1234 5678 9012 3456

display as : **** **** **** 3456

In [108…
```
def mask_credit_card(card_number):
    parts = card_number.split()
    masked = ["****"] * (len(parts) - 1) + [parts[-1]]
    return " ".join(masked)


card_number=(input("Enter the card number: "))
print("Masked card number:", mask_credit_card(card_number))
```

```
Masked card number: **** **** **** **** 25478
```

11) : Checking if the two strings are Anagram or not.

s1 = decimal and s2 = medical are Anagram

In [96]:
```
def are_anagrams(s1, s2):
    return sorted(s1.lower()) == sorted(s2.lower())

s1 = "mahekk"
s2 = "jahekk"
if are_anagrams(s1, s2):
    print(f'"{s1}" and "{s2}" are anagrams.')
else:
    print(f'"{s1}" and "{s2}" are not anagrams.')
```

```
"mahekk" and "jahekk" are not anagrams.
```

12) : Rearrange the given string. First lowercase then uppercase alphabets.

input : EHlsarwiwhtwMV

output : lsarwiwhtwEHMV

In [91]:
```
def rearrange_string(s):
    lowercase = ''.join([char for char in s if char.islower()])
    uppercase = ''.join([char for char in s if char.isupper()])
    return lowercase + uppercase

input_str = "EHlsarwiwhtwMV"
```

```python
output_str = rearrange_string(input_str)
print("Rearranged string:",output_str)
```

```
Rearranged string: lsarwiwhtwEHMV
```

In [ ]: