** **

**The assignment report**

CS111 with the Eng Samar

Name: Anas Adel Ahmed ID: 20220070

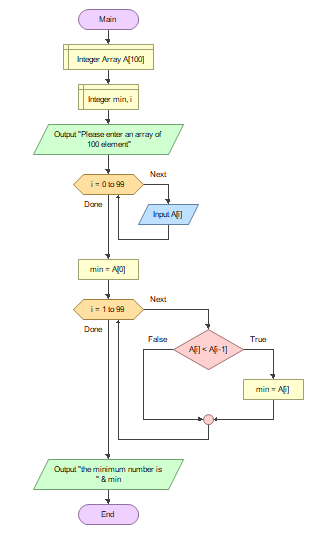
E-mail: [anasadel898@gmail.com](mailto:anasadel898@gmail.com)

Name: Nour El-deen Mohamed ID: 20220365

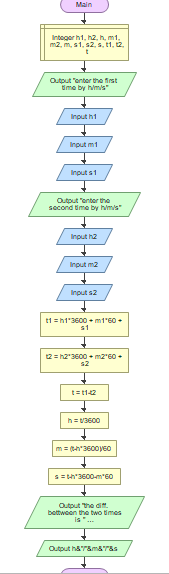
E-mail: [noureldeen2005@gmail.com](mailto:noureldeen2005@gmail.com)

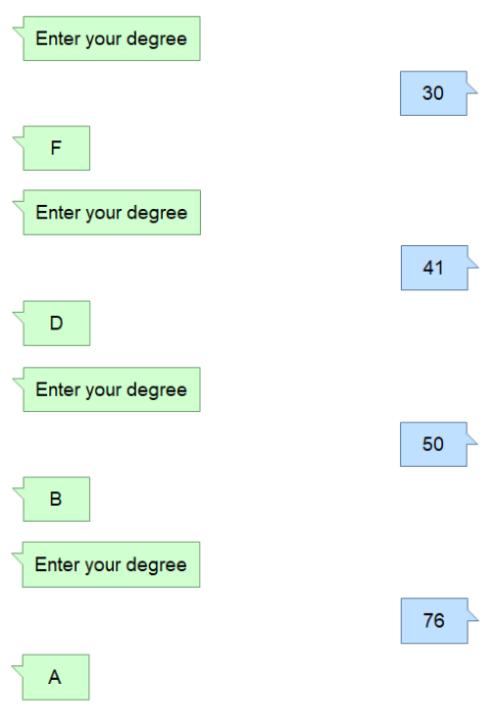
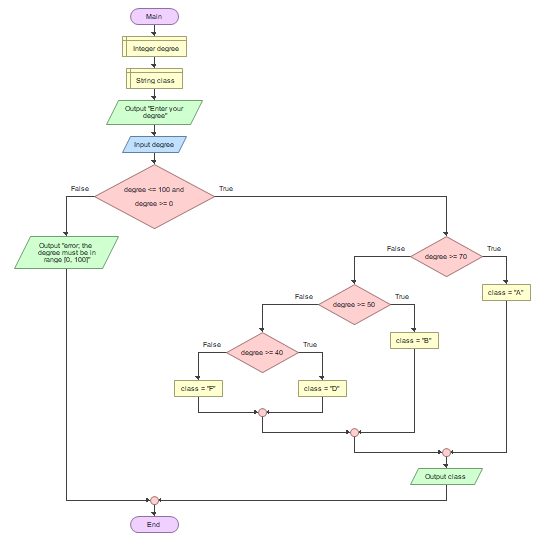
**Part 1**

**Q1**: The flowchart is below.

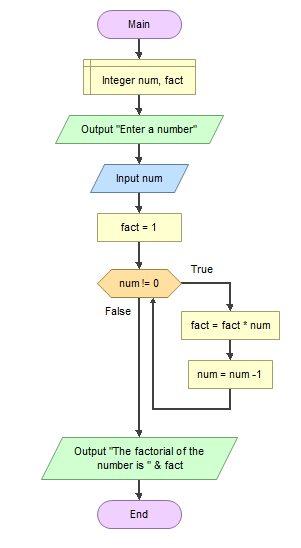


**Q2**: The python code & the flowchart are below.

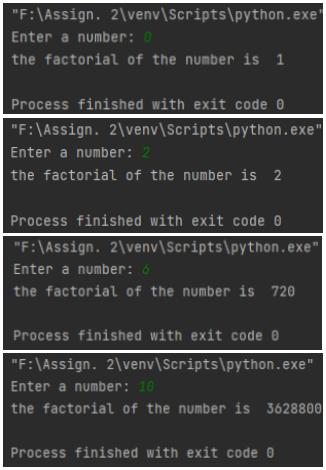
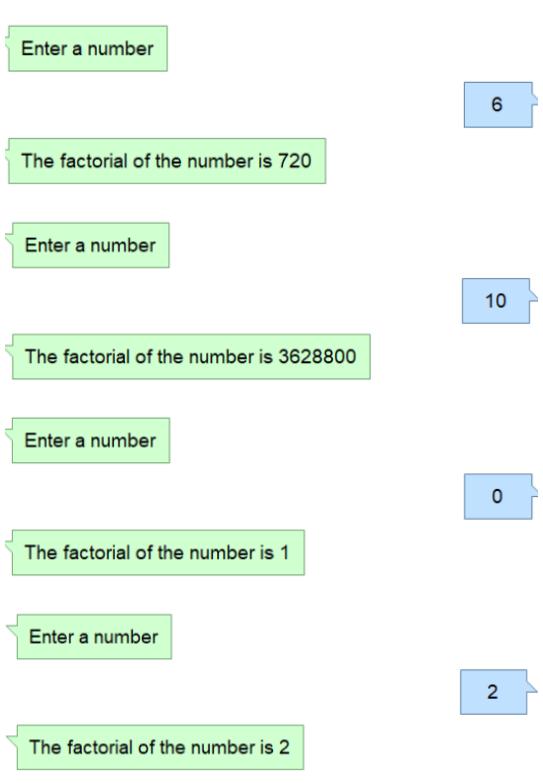
 print("enter the first time by h/m/s")  
h1 = int(input())  
m1 = int(input())  
s1 = int(input())  
print("enter the second time by h/m/s")  
h2 = int(input())  
m2 = int(input())  
s2 = int(input())  
t1 = h1 \* 3600 + m1 \* 60 + s1  
t2 = h2 \* 3600 + m2 \* 60 + s2  
t = t1 - t2  
h = int(t / 3600)  
m = int((t - h \* 3600) / 60)  
s = int(t - h \* 3600 - m \* 60)  
print("the diff. between the two times is ", end='', flush=True)  
print(str(h) + "/" + str(m) + "/" + str(s))

**Q3**: The flowchart and the test are below.

**Q4**: The python code & the flowchart are below.

num = int(input("Enter a number: "))  
fact = 1  
while num != 0:  
 fact = fact \* num  
 num -= 1  
print("the factorial of the number is ", fact)

**Cont. Q4**: The tests are below.



**Q5**: The python code and the trace are below.

word = input("Enter a word: ")  
list\_word = list(word)  
len = len(word)  
for i in range(0, len):  
 list\_word[i] = word[len-1-i]  
rev\_word = ''.join(list\_word)  
if word == rev\_word:  
 print("this word is palindrome")  
print(rev\_word)

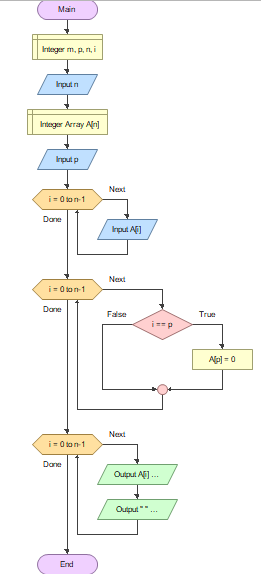
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| line | word | list\_word | len | i | rev\_word |
| 1 | "xyzzyx" | ? | ? | ? | ? |
| 2 | = | ['x','y','z','z','y','x'] | ? | ? | ? |
| 3 | = | = | 6 | ? | ? |
| 4 | = | = | = | 0 | ? |
| 5 | = | = | = | 1 | ? |
| 5 | = | = | = | 2 | ? |
| 5 | = | = | = | 3 | ? |
| 5 | = | = | = | 4 | ? |
| 5 | = | = | = | 5 | ? |
| 6 | = | = | = | = | "xyzzyx" |
| 8 | Output: | "This word is palindrome." | | |  |
| 9 | Output: | rev\_word |  |  |  |

**Q6**: The python code is below.

A = [0] \* 50  
print("Enter a list")  
for g in range(0, 50):  
 A[g] = input()  
B = list(A)  
i = 0  
while i < len(A):  
 A[i] = A[i+1]  
 A[i+1] = B[i]  
 i += 2  
print(A)

**Q7**: The python code is below.

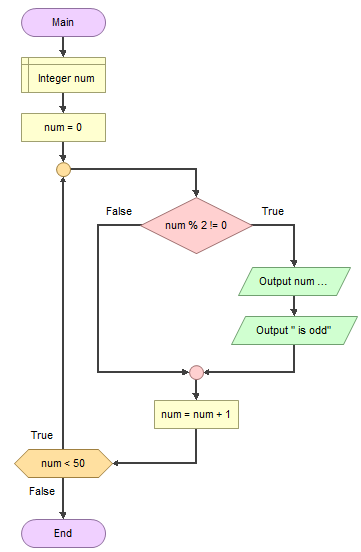
A = [0] \* 4  
X = [0] \* 4  
print("Enter a list")  
for i in range(0, 4):  
 A[i] = input()  
for i in range(0, 4):  
 X[i] = A[3-i]  
print(X)

**Q8**: The flowchart is here 🡪

**Q9**: Yes, there is a big difference between both of them, and if an algorithm was not written with enough details the algorithm will be ambiguous for the computer.

**Q10**: Primitives are precisely defined, and each one has its own meaning. And they have low level of ambiguity, so by using them in an algorithm, it will be built by precise, clear and unambiguous tools or steps.

**Q11**: The flowchart is below.



**Q12**: The algorithm is in the sol. folder.

num = 100  
while (num > 0):  
 print(num)  
 num -= 1

**Q13**: The algorithm is in the sol. folder.

num = int(input("Enter a number: "))  
if num > 1:  
 factors = []  
 for i in range(1, num+1):  
 if num % i == 0:  
 factors.append(i)  
 print("The positive factors of the number are ", factors)  
elif num == 1 or -1:  
 print("the positive factor of the number is ", num)  
elif num == 0:   
 print("The positive factors of the number are ", factors)  
 print("The factors of zero are all the numbers")  
else:  
 num = -num  
 factors = []  
 for i in range(1, num+1):  
 if num % i == 0:  
 factors.append(i)

**Q14**: The solution is below S E N D

+ M O R E

M O N E Y

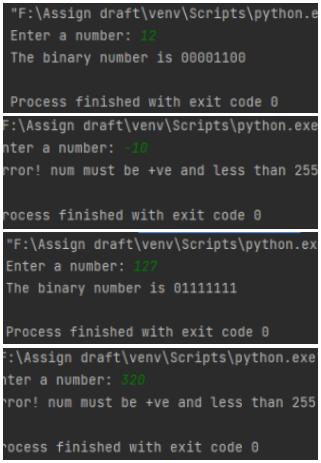
* Since that every letter represents only one digit
* Therefor the addition of a two digits cannot be more than 18 and if there is a 1 from the previous addition then the max sum of S & M is 19
* There for **M = 1**
* Since that M = 1 and the sum of S & M is greater than 9 and the max number of any letter is 9
* Therefor 1 + S > 9 and S < 10 then S > 8 and S < 10
* Therefor **S = 9** and **O = Zero**
* Since O = Zero and that each letter represents a different digit
* Therefor E = N is refused and E + 1 = N is accepted
* Therefor N + R = E + 10 or N + R = E + 9
* Therefor if we substitute N by E + 1 , R = 9 ref. bec. S=9 then **R = 8**
* Therefor D + E = Y + 10 and previously E + 1 = N
* Since that 0 ,1 ,8 and 9 has been taken , therefor the domain [2,7]
* Therefor the only numbers that will verify the previous conclusion and the previous two equations are **D = 7** , **E = 5** , **N = 6** and **Y = 2**

**Q15**: Yes, it is correct since that if ( y % x = 0 ) that means that the x is a divisor of the y and vice versa.

**Q16**: The pseudo code is below

|  |
| --- |
| 1. Input(Text)  2. result = ""  3. for i in range(len(text)):  char = text[i]  if char == "z" or char == "Z":  s = chr(ord(char) - 25)  else:  s = chr(ord(char) + 1)  result += s  4. print(result) |

**Q17**: The pseudo code and tests are below.



1. dec = int(input("Enter a number: "))

2. bin = ""

3. if dec < 256 and dec >= 0:

for i in range(8):

x = 7 - i

if dec / (2 \*\* x) >= 1:

bin += "1"

dec -= 2 \*\* x

else:

bin += "0"

print("The binary number is " + bin)

else:

print("Error! num must be +ve and

less than 255")

**Q18**: The pseudo code is below.

1. pi= 0.0

2. n = int(input("Enter a number: "))

3. For i in range(1, n, 4):

pi += 4/ i

pi -= 4/ (i + 2)

4. print(pi)

**Part 2**

**Q19**:

a) 0 🡪 +ve 101 🡪 e = 1 bin = 1.001

The dec. number equals 1.125

b) 1 🡪 -ve 010 🡪 e = -2 bin = 0.0011

The dec. number equals 0.1875

c) 1 🡪 -ve 000 🡪 e = -4 bin = .00001111

The dec. number equals 0.05859375

**Q20**:

1. **1 01111101 01100001010001111010111**

Num = (-1)s  \* 2E -127  \* 1.M

= (-1)1 \* 2125-127  \* 1.M = - 2-2  \* 1.M

Approximated to**: -0.345**

2. **0 10000101 1111101000000…**

Num = (-1)0 \* 2133-127  \* 1.1111101

= 26 \* 1.1111101 = 1111110.1 = **126.5**

**Q21**: Since that A1+ A1+ A1- A2 = - (A2- A1+ A1+ A1) and that this

factor is in the numerator and the denominator , So the Zero

factor was canceled up and down so there is no DIV/0 ERROR.

**Q22**: There is overflow at operations B, C, E

By performing the operations we will find that the result needs 5 bit to be stored but we only have 4 bits, that’s why overflow occurred.

B: **1100 + 0100 = (1)0000 = 0, the result was supposed to be 16**

**12 + 4 = 16**

C:  **1100 + 1100 = (1)1000 = 8, the result was supposed to be 24**

**12 + 12 = 24**

E:  **1011 + 1100 = (1)0111 = 7, the result was supposed to be 23**

**11 + 12 = 23**

**Q23**: The message says “Computer Science”.

**Q24**: To represent the digits and elements that couldn’t be represented by the ASCII. The ASCII only represented 128 elements, which was obviously not enough to represent other languages letters such as Arabic and Chinese languages.

**Points of Comparison: 1. Size 2. ASCII == Unicode?**

1. **Size:** It’s obvious that the Unicode takes more space than ASCII, since ASCII takes only up to 7 bits to represent an element. The Unicode encodes more than a 150 written scripts, in addition to emojis. This of course makes the Unicode consumes more space than ASCII
2. **ASCII == Unicode?** The first 128 elements from the Unicode are the same 128 elements of the whole ASCII, so we can say that the ASCII is subset from Unicode.

**Part 3**

**Q25**: b. is dynamically typed

c. is interpreted

d. uses type inference

**Q26**: 1. width / 2 = 7.5 <class ‘float’>

2. x / 2 = - 1.5 <class ‘float’>

3. width / 2.0 = 7.5 <class ‘float’>

4. height / 3 = 5 <class ‘int’>

5. 1+2\*5 = 11 <class ‘int’>

**Q27**: The python code is below.

num = input("Enter a number: ")  
sum = 0  
for i in range(len(num)):  
 s = int(num[i])  
 sum += s  
print(sum)

**Q28**: The updated version is below.

num = input("Enter a number: ")  
result = 0  
  
# Calc. the sum of cubes of the number's digits  
for i in range(len(num)):  
 s = int(num[i])  
 result += s \*\* 3  
  
# Comparing the number with the cubes of its digits  
if int(num) == result:  
 print("The given number is Armstrong")  
else:  
 print("The given number is not Armstrong")

**Q29**: The python code is below.

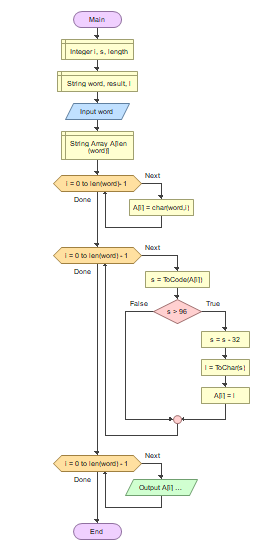
start = int(input("Enter the starting number: "))  
end = int(input("Enter the ending number: "))  
nums = []  
for i in range(start, end + 1):  
 if i % 9 == 0 and i % 4 != 0:  
 nums.append(i)  
print("The nums that are divisible by 9 and not by 4 are", nums)

**Q30**: The python code is below.

x = int(input("Enter the base: "))  
y = int(input("Enter the power: "))  
power = 1  
  
for i in range(y):  
 power = power \* x  
   
print(power)

**Q31**: The python code is below.

print("enter three numbers below")  
x = int(input())  
y = int(input())  
z = int(input())  
  
if x - y == z:  
 print(str(x) + " - " + str(y) + " = " + str(z))  
elif x + y == z:  
 print(str(x) + " + " + str(y) + " = " + str(z))  
elif x \* y == z:  
 print(str(x) + " \* " + str(y) + " = " + str(z))  
elif x / y == z:  
 print(str(x) + " / " + str(y) + " = " + str(z))  
elif x \*\* y == z:  
 print(str(x) + " \*\* " + str(y) + " = " + str(z))  
else:  
 print("There is no any operations between these numbers")

**Q32**:

1. input (word)

2. A = word.split()

3. for i = 0 to i = len(word):

s = ord(A[i])

if s > 96:

s = s - 32

l = chr(s)

A[i] = l

4. word = “”.join(A)

5. print (word)

**Q33**: The correct choice is

3) By adding big\_nums = [] on line 2

**Q34**:

1. The syntax error is like a grammatical error in programming language:

x = int(input(Enter a number: )) 🡪 There should be quotation marks between the red statement

y = int(input(Enter another number: )) 🡪 Same as above

if x >= y 🡪 There should be a colon ( : ) after the if condition

print(x + y)

2. The run-time error occurs after the program is successfully compiled, which means that it has no syntax errors Such as:

a) Division by zero

b) Performing infinite loop

c) Using an identifier that has not been identified

d) Accessing a file that doesn’t exist

3. Although logical errors are errors that has neither syntax nor run-time errors, but it doesn’t perform or show the exact right output

Ex: x = int(input("Enter a number: "))

y = int(input("Enter another number: "))

# Calculating the sum of x & y sumValue = x - y 🡪Here we wrote by mistake (-) instead of (+), which is not wrong as a syntax, but it doesn’t perform the right task print(sumValue)

**Q35**: The python code is below.

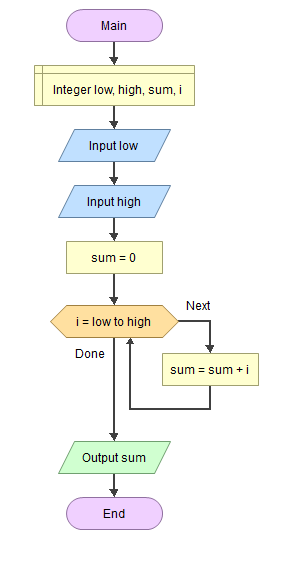
string = input("Enter any type of text: ")  
i = int(input("Enter a number less than the length of the text: "))  
string = string.replace(string[i], "")  
print(string)

**Q36**: The python code is below.

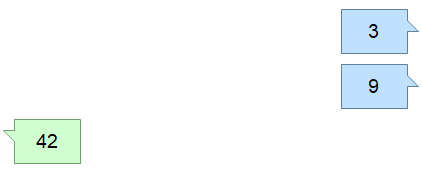
word = input("Enter a word: ")  
  
reverse = word[::-1]  
  
if (reverse == word):  
 print("The word is Palindrome")  
else:  
 print("Not Palindrome")

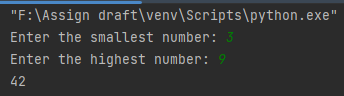
**Q37**: The python code is below.

A = []  
length = int(input("Enter the length of the array: "))  
print("Enter the array elements below:")  
for i in range(length):  
 x = input("")  
 A.append(x)  
print("Enter 3 numbers below:")  
m = int(input())  
p = int(input())  
n = int(input())  
if m <= p <= n:  
 A.remove(A[p])  
print(A)

**Q39**: The python code, the flowchart and the tests are below.

Low = input("Enter the lowest "))  
high = input("Enter the highest number:")  
sum = 0  
for i in range(int(low),int(high + 1)):  
 sum += i  
print(sum)





**Q40**: The python code is below.

num = int(input("Enter the max num: "))  
a = 0  
b = 1  
if (num == 1):  
 print(a)  
elif (num == 2):  
 print(a)  
 print(b)  
else:  
 print(a)  
 print(b)  
 for i in range(2, num, 1):  
 c = a + b  
 a = b  
 b = c  
 print(c)