** **

**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)

**1. Bits and Their Storage**

1) a. output 🡪 0 b. output 🡪 1 c. output 🡪 1

2) a. AND gate b. XOR gate

3) We are going to connect the inputs to two circuits.

The first one will take both inputs into an OR gate,

and the second one will take them to an NAND gate,

and the taking both outputs into an And gate.

**2. Main Memory**

4) a. MSB value is 192 , LSB value is 13

b. MSB value is 96 , LSB value is 7

c. MSB value is 128 , LSB value is 1

d. MSB value is 128 , LSB value is 0

**5)** Address: 00 | 01 | 02 | 03

Content: 02 | 53 | 53 | 53

**6)** If 2 hexadecimal digits, then the max number of cells will equal *255* cells, and the last address will be FF (in hex)

If 4 hex digits, then the max number of cells will equal *65,535* cells, and the last address will be FFFF (in hex)

7) a. Flash Drives b. CDs c. Hard disks d. CDs

e. Hard disks f. Flash Drives (To access data very quick)

**8)** Each up and down is equal to one, and each straight line will equal 0 until a change happens:

9) approximately 5,083 bitmap photos

10) approximately 1.572 MB required to store a photo

11) When ASCII is used it will use 1.4 MB of the storage ,

Else if UTF-16 is used it will use 2.8 MB .

**12)** a. capacity = cylinders \* tracks \* sectors \* bps \* 2( 2 faces)= 569.98 gigabytes

b. time = (1 \* 1 \* 60)/ 5400 = 0.011 s = 11.11 ms

c. avg. latency = time for 1 cycle / 2 = 11.11/2 = 5.55 ms

d. time to transfer to 1 sector = time for 1 cycle/ no. of sectors = 11.11/ 600 = 0.0185 ms

e. total time = access time + time to transfer 20400 bytes = (seek time + avg. latency) + (time transfer 1 byte \* 20400) = (11 + 5.55) + [(0.0185 / 5400) \* 20400] = 16.62 ms

13) a. sector, track, cylinder

**14)** c Exabyte

15) c. 0.008333 sec

**16)**

17) magnetic disks got more storage and it is cheaper than SSD.

**4. Image representation**

19) a. 0.131 MB b. 1.04 MB c. 3.145 MB

**20) a.** The vector image is resolution independent, it doesn’t have to be in a particular shape, and it can be easily converted into bitmap image.

**b.** Because GIF is lossless compression, while JPEG is lossy compression.

**c.** JPEG exploits the characteristics of human vision, eliminating or reducing data to which the eye is less sensitive. Slight changes in color are not perceived well by the human eye, while slight changes in intensity (light and dark) are, as in JPEG.

**d.** If I want to save space, then I would use the JPEG, because it will consume less space, but if I want to preserve the quality, I would use GIF, it won’t save the same space as JPEG do, but it will keep the quality good.

21) a. The resolution is measured by how many pixels in an inch

b. The storage needed respectively is 60KB, 480KB, 1440KB

c. PNG gives you a true color, since it have an extra byte for

representing transparency, but it requires large storage.

d. lossy compression can be done to all the data, while

lossless compression can’t

**5. Machine Language:**

**22)** The program prints the letter that is in R3 to the screen.

**23)** 00 11A0

02 2280

04 8212

06 2000

08 B20C

0A C000

0C 207F

0E 8301

10 33A0

**24)**

**25)** a. Move the bit pattern found in register 3 to register 4

b. AND the bit patterns in reg. 2 and reg. 3, and put the result in reg. 0

c. If reg. 2 == reg. 0 jump to location 88 in memory

d. load to reg. 3 the bit pattern 45

**26) a.** 27A5 b. 17A5 c. 5456 d. 7456

**27)** a. the bit pattern registered in R4 is (C0)

b. the bit pattern registered in R1 is (0B)

**28) A)** (C0)h or (1100 0000)bin B) it will be in stages (03 04) in memory C) (00)hex

**29)** 00 2100 // load 00 to R1

02 31F0 // Storing R1 in variable x (F0 in memory)

04 2004 // load 04 to R0

06 B110 // If R1 == R0(4) jump to 10 in memory

08 2201 // load 01 to R

0A 5112 // R1 = R1 + R2(01)

0C B006 // unconditional jump to 0

0E 31F0 // print X (F0)

10 C000 // end

**6.Python:**

**30)** def encrypt(text):  
 k = 1  
 result = ""  
 for i in range(len(text)):  
 char = text[i]  
   
 if char.isupper():  
 s = chr(((ord(char) - 65 + k) % 26) + 65)  
 result += s  
   
 elif char.islower():  
 s = chr(((ord(char) - 97 + k) % 26) + 97)  
 result += s  
 print(result)  
 return result  
  
def decrypt(text):  
 k = -1  
 result\_decrypt = ""  
 for i in range(len(text)):  
 char = text[i]  
   
 if char.isupper():  
 s = chr(((ord(char) - 65 + k) % 26) + 65)  
 result\_decrypt += s  
   
 elif char.islower():  
 s = chr(((ord(char) - 97 + k) % 26) + 97)  
 result\_decrypt += s  
 print(result\_decrypt)  
   
   
encrypt("I love Python Programming.")  
decrypt("JmpwfQzuipoQsphsbnnjoh")

31) The python code is below

def equal\_lists(lst1, lst2):  
 for i in lst1:  
 for l in lst2:  
 x = False  
 if i == l:  
 x = True  
 break  
 if not x:  
 return False  
 break  
 else:  
 return True

**32)** # Open the file to save the data in, in write mode  
f1 = open(input("Enter the file you want to save the output in: "), "w")  
  
# Open the input file you want to get the data from  
with open(input("Enter the file to take the data from: "), "r") as myfile:  
 data = myfile.read()  
  
  
data\_1 = data[::-1]  
f1.write(data\_1)  
  
f1.close()

33) The python code is below.

def sort(lst):  
 x = 0  
 for i in range(len(lst) - 1):  
 x = 1  
 if lst[i] >= lst[i + 1]:  
 x = 0  
 break  
 if x == 0:  
 for i in range(len(lst) - 1):  
 x = -1  
 if lst[i] <= lst[i + 1]:  
 x = 0  
 break  
 return x

**34)**

def is\_win():  
 global winner  
 a,b,c,d,e,f,g,h=[1,5,9],[1,6,8],[2,4,9],[2,5,8],[2,6,7],[3,5,7],[4,3,8],[4,5,6]  
 if ((all(x in x\_list for x in a)) or (all(x in x\_list for x in b))or (all(x in x\_list for x in c))or(all(x in x\_list for x in d))or (all(x in x\_list for x in e))or (all(x in x\_list for x in f))or (all(x in x\_list for x in g))or (all(x in x\_list for x in h))):  
 winner = True  
 print("X has won!")  
 elif ((all(x in y\_list for x in a)) or (all(x in y\_list for x in b))or (all(x in y\_list for x in c))or(all(x in y\_list for x in d))or (all(x in y\_list for x in e))or (all(x in y\_list for x in f))or (all(x in y\_list for x in g))or (all(x in y\_list for x in h))):  
 winner = True  
 print("Y has won!")  
  
nums = [1, 2, 3, 4, 5, 6, 7, 8, 9]  
x\_list = []  
y\_list = []  
winner = False  
turn = 1  
  
while (nums != [] and winner == False):  
 print(nums)  
 print("\n")  
 #if turn is even, then x choose  
 if (turn % 2 == 0):  
 x = int(input("Player X choose from 1 to 9 & not chosen: "))  
 if x in nums:  
 x\_list.append(x)  
 nums.remove(x)  
 else:  
 x = int(input("The num is taken, enter another one: "))  
 x\_list.append(x)  
 nums.remove(x)  
 is\_win()  
   
 #if turn is odd, then y choose  
 else:  
 y = int(input("Player Y choose from 1 to 9 & not chosen: "))  
 if y in nums:  
 y\_list.append(y)  
 nums.remove(y)  
 else:  
 y = int(input("The num is taken, enter another one: "))  
 y\_list.append(y)  
 nums.remove(y)  
 is\_win()  
 turn += 1  
if (nums == [] and winner == False):  
 print("It is a tie")

35) The python code is below.

print("Subtract a square game")  
num = 50  
turn = 1  
while num > 0:  
 print("Number = ", num)  
 if turn == 1:  
 print("Player 1 turn")  
 x = int(input("Enter a squared number less than 25: "))  
 turn = 2  
 if x in [1, 4, 9, 16]:  
 num -= x  
  
 else:  
 print("number must be squared and less than 25")  
 turn = 1  
  
 else:  
 print("Player 2 turn")  
 x = int(input("Enter a squared number less than 25: "))  
 turn = 1  
 if x in [1, 4, 9, 16]:  
 num -= x  
 else:  
 print("number must be squared and less than 25")  
 turn = 2  
  
if turn == 2:  
 print("Player 1 has won")  
else:  
 print("Player 2 has won")

**36)**

f2 = open(input("Enter the file that I will paste in: "), "w")  
  
with open(input("Enter the file that I will copy from: "), "r") as f1:  
 data = f1.read()  
  
f2.write(data)  
f2.close()

37) The python code is below.

read\_file = open("file.txt", "r")  
file = read\_file.read().split()  
l = len(file)  
for i in range(l-1):  
 temp = file[i]  
 file[i] = file[l-1-i]  
 file[l-1-i] = temp  
print(file)  
read\_file.close()

**38)**

file = open(input("Enter the file: "), "r")  
data = file.read()  
lines = len(data.split("\n")) #--> Each new line will be considered as split, and it will count each split  
words = len(data.split())  
chars = len(data)  
print("LINES = ", lines)  
print("WORDS = ", words)  
print("CHARS = ", chars)

39) The python code is below.

egy\_cities = {"Cairo": 7000000, "Alex": 4000000, "Giza": 250000, "Port Said": 500000,  
 "suez": 450000, "Luxor": 400000, "Asyut": 400000, "Tanta": 400000,  
 "Fayom": 300000, "Ismailia": 250000}  
  
  
print("'a' to get the population of a city\n"  
 "'b' to add a city's population\n"  
 "'c' to list all the city's populations")  
choice = input("chose (a, b, c): ")  
if choice == "a":  
 city = input("Enter the city: ")  
 print("The population of the city is", egy\_cities[city])  
elif choice == "b":  
 city = input("Enter the city: ")  
 pop = input("Enter the population")  
 egy\_cities[city] = pop  
elif choice == "c":  
 for i in egy\_cities:  
 print(i, ": ", egy\_cities[i])  
else:  
 print("a, b, c only allowed")

**40)**