**DE NOBILI SCHOOL KORADIH**

**SUMMER VACATION HOMEWORK(2025-26)**

**Class – 12 subject :- Computer Application**

**Question 1:**

1. Find the complement of XY’Z + XY + YZ’
2. State the principle of duality. Write the dual of (P + Q’).R.1 = PR + (Q’.R)
3. Convert the following expression into its canonical POS form : F(A,B) = (A+B).A’
4. i) **Reduce the Boolean function**  F(A,B,C,D) =(0,2,4,6,8,9,10,11,14) By using four variable turn up map showing the various groups (that is octal quad and Pairs)

ii) Draw the logic diagraph for the reduced expression assume that the variables and their complements are available as inputs.

1. **Reduce the Boolean function**

F(A,B,C,D)=π(0,2,4,6,8,9,10,11,14)

**i)** Use a four‑variable Karnaugh map to simplify F, showing all octets, quads, and pairs.  
**ii)** Draw the logic diagram for the minimized expression, assuming A,B,C,D and their complements are available as inputs.

1. **Reduce the Boolean function**

F(A,B,C,D)=π(0,1,3,5,7,8,10,12,13,15)

**i)** Use a four‑variable Karnaugh map to simplify F, showing all octets, quads, and pairs.  
**ii)** Draw the logic diagram for the minimized expression, assuming A,B,C,D and their complements are available as inputs.

1. **Reduce the Boolean function**

F(A,B,C,D)=π(1,2,4,5,9,11,12,13,14,15)

**i)** Use a four‑variable Karnaugh map to simplify F, showing all octets, quads, and pairs.  
**ii)** Draw the logic diagram for the minimized expression, assuming A,B,C,D and their complements are available as inputs.

1. **Reduce the Boolean function**

F(A,B,C,D)=π(0,2,3,6,7,9,10,11,12,14)

**i)** Use a four‑variable Karnaugh map to simplify F, showing all octets, quads, and pairs.  
**ii)** Draw the logic diagram for the minimized expression, assuming A,B,C,D and their complements are available as inputs.

1. **Reduce the Boolean function**

F(A,B,C,D)=Σ(0,1,2,5,7,8,9,10,13,15)

**i)** Use a four‑variable Karnaugh map to simplify F, showing all octets, quads, and pairs.  
**ii)** Draw the logic diagram for the minimized expression, assuming A,B,C,D and their complements are available as inputs.

1. **Reduce the Boolean function**

F(A,B,C,D)=Σ(1,3,4,6,7,11,12,13,14,15)

**i)** Use a four‑variable Karnaugh map to simplify F, showing all octets, quads, and pairs.  
**ii)** Draw the logic diagram for the minimized expression, assuming A,B,C,D and their complements are available as inputs.

1. **Reduce the Boolean function**

F(A,B,C,D)=Σ(2,3,4,5,6,7,8,10,11,14)

**i)** Use a four‑variable Karnaugh map to simplify F, showing all octets, quads, and pairs.  
**ii)** Draw the logic diagram for the minimized expression, assuming A,B,C,D and their complements are available as inputs.

1. **Reduce the Boolean function**

F(A,B,C,D)=Σ(0,2,4,6,8,9,11,12,14,15)

**i)** Use a four‑variable Karnaugh map to simplify F, showing all octets, quads, and pairs.  
**ii)** Draw the logic diagram for the minimized expression, assuming A,B,C,D and their complements are available as inputs.

**Question 2.**

Design a class **ArmNum** to check if a given number is an Armstrong number or not.  
[A number is said to be Armstrong if sum of its digits raised to the power of length of the number is equal to the number]   
Example:  
371 = 3³ + 7³ + 1³  
1634 = 1⁴ + 6⁴ + 3⁴ + 4⁴  
54748 = 5⁵ + 4⁵ + 7⁵ + 4⁵ + 8⁵  
Thus, 371, 1634 and 54748 are all examples of Armstrong numbers.  
Some of the members of the class are given below:

* Class name: **ArmNum**
* Data members/instance variables:
  + n: to store the number
  + l: to store the length of the number
* Methods/Member functions:
  + ArmNum(int nn): parameterized constructor to initialize the data member n = nn
  + int sum\_pow(int i): returns the sum of each digit raised to the power of the length of the number using recursion
  + void isArmstrong(): checks whether the given number is an Armstrong number by invoking sum\_pow() and displays the result  
    Specify the class **ArmNum**, giving details of the constructor, int sum\_pow(int) and void isArmstrong(). Define a main() function to create an object and call the functions accordingly.

**Question 3**

**Write a class program. Class name:** WordSorter

* **Data member:** String[] words
* **Member functions:**
  + WordSorter(): to initialize the array words with default values // default constructor
  + void input(): to accept a list of words
  + void sortWords(): to sort the words in alphabetical order
* **Task:** Display the sorted list of words.

### **Question 4**

**Write a class program. Class name:** StringManipulator

* **Data member:** String text
* **Member functions:**
  + StringManipulator(): to initialize text with an empty string // default constructor
  + void input(): to accept a string
  + void reverseString(): to reverse the string
* **Task:** Display the reversed string.

### **Question 5**

**Write a class program. Class name:** AnagramChecker

* **Data members:** String str1, str2
* **Member functions:**
  + AnagramChecker(): to initialize str1 and str2 with empty strings // default constructor
  + void input(): to accept two strings
  + void checkAnagram(): to check whether the two strings are anagrams
* **Task:** Display an appropriate message indicating whether the strings are anagrams.

### **Question 6**

**Write a class program. Class name:** PalindromeChecker

* **Data member:** String word
* **Member functions:**
  + PalindromeChecker(): to initialize word with an empty string // default constructor
  + void input(): to accept a word
  + void isPalindrome(): to check whether the word is a palindrome
* **Task:** Display an appropriate message indicating whether the word is a palindrome.

### **Question 7**

**Write a class program. Class name:** FrequencyCounter

* **Data member:** String sentence
* **Member functions:**
  + FrequencyCounter(): to initialize sentence with an empty string // default constructor
  + void input(): to accept a sentence
  + void countFrequency(): to count and display the frequency of each character
* **Task:** Display the frequency of each character in the sentence.

**Question 8**

A class **Rearrange** has been defined to modify a word by bringing all the vowels in the word at the beginning followed by the consonants.   
Example:

ORIGINAL → REARRANGED

ORIGINAL → OIIARGNL

Some of the members of the class are given below:

* Class name: **Rearrange**
* Data members/instance variables:
  + wrd: to store a word
  + newwrd: to store the rearranged word
* Member functions/methods:
  + Rearrange(): default constructor
  + void readword(): to accept the word in UPPERCASE
  + void freq\_vow\_con(): finds and displays the frequency of vowels and consonants in the word
  + void arrange(): rearranges the word by bringing the vowels at the beginning followed by consonants
  + void display(): displays the original word along with the rearranged word  
    Specify the class **Rearrange**, giving the details of the constructor, void readword(), void freq\_vow\_con(), void arrange() and void display(). Define the main() function to create an object and call the functions accordingly.

**Question 9**

A superclass **Record** contains names and marks of the students in two different single‑dimensional arrays. Define a subclass **Highest** to display the names of the students obtaining the highest mark.   
The details of the members of both classes are given below:

**Class name:** Record

* Data members/instance variables:
  + n[]: array to store names
  + m[]: array to store marks
  + size: to store the number of students
* Member functions/methods:
  + Record(int cap): parameterized constructor to initialize size = cap
  + void readarray(): to enter elements in both arrays
  + void display(): displays the array elements

**Class name:** Highest

* Data member/instance variable:
  + ind: to store the index
* Member functions/methods:
  + Highest(int cap): parameterized constructor to call super and initialize arrays
  + void find(): finds the index of the student obtaining the highest mark and assigns it to ind
  + void display(): displays the array elements along with the name(s) and mark(s) of the student(s) who have obtained the highest mark

Assume that the superclass **Record** has been defined. Using the concept of inheritance, specify the class **Highest**, giving the details of the constructor, void find() and void display(). The superclass, main() function and algorithm need **NOT** be written.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*