Assignment-04

Hrudaya
TA40935

Problem 01:-

Let $\tilde{a}$ and $\tilde{b}$ be polynomials in $GF(2)[x]$, and let $\hat{q}$ and $\hat{r}$ be the corresponding unique polynomials in $GF(2)[x]$ such that

$$\tilde{a} = \tilde{b}\tilde{q} + \tilde{r}$$

where $\tilde{r} = 0$ or $\deg(\tilde{r}) < \deg(\tilde{b})$ construct in pseudocode two algorithms $QUO(\tilde{a}, \tilde{b})$ and $REM(\hat{a}, \tilde{b})$ which respectively compute $\hat{q}$ and $\tilde{r}$

solution :.

Here $\deg(\tilde{a})$ we will consider the degree of polyminomial $\tilde{a}$ which gives highest power'x' with a non zero coeficient

Ily $\deg(\tilde{b})$ consider the degree of polynomial $\tilde{b}$ which gives highest power'x' with non zero coefficient

Firstly to give a rough idea in order to implement $QUO(\tilde{a}, \tilde{b})$, we will loop till $\deg(a) >= \deg(b)$ where $\deg(a)$ is subracted by $\deg(b)$ which we stored in variable 'z' and by this 'z' we left shift 'b'.

The quotient is what we get as xor of x with z ie x^z. we initially make $q=0$ and we keep on adding $q+x^z$ until $deg(a) >= deg(b)$

```
q=0
while deg(a) >=deg(b):
    z=deg(a) - deg(b)
    q=q+x^z
     a=a+(b<<z)
return q
```

In order to implement remainder algorithm REM(a,b) we will loop till $deg(\hat{a}) >= deg(\bar{b})$ where $deg(\hat{a})$ is subracted by $deg(\bar{b})$ which we store in a variable 'z' and we repeatedly so this 'z' by left shift of 'b', which we keep on adding a with left shift of b and 'd'

Hence return 'a' which is remainder of a and b

```
REM(a,b):
    while deg(a) >=deg(b):
        z =deg(a)-deg(b)
        a=ar(b<<z)
        return a
```

pseudo code

```
QUO(ã, b̃):

    q = 0
    while deg(â) >= deg(b̂):
        z = deg(â) - deg(b̂)
        q = q + x^z
        â = ã + (b̃ << z)
    return q

REM(â, b̃):
    while deg(ã) >= deg(b̂):
        z = deg(ã) - deg(b̂)
        ã = â + (b̃ << z)
    return ã
```

Problem-02

Using the above algorithmic procedures $QUO(\tilde{a},\tilde{b})$ and $REM(\tilde{a},\tilde{b})$, construct in pseudocode an algorithmic procedure. INVERSE(a) which computes the inverse $\bar{a}^1$ of a in $GF(2^n)$ provided a to

solution

Here given n is degree of 'a'

In order to construct field x^n+1 is a irreducible polynomial in $GF(2^n)$

In order to construct a pseudo code for INVERSE(a) the quotient and remainder functions are used from above

The QUO function is used to perform quotientgie b on a QUO(b, a)

The REM function is used to check if a is invertible or not.

If a is not invertible
    return '0'
  if REM(b, a) == 0 :
      return 0

If a is invertible we have to perform euclideor algorithm to find inverse

The algorithm maintains two sequence of polynomial u-i and v-i such that u_i = a * v_i mod b, at each step, it computes the next two terms in a sequence by using previous two terms and the quotient of division,

when the degree of r_i reaches zero it stops and returns last two term in the sequence.
Hence it is the inverse of 'a'

INVERSE$_E(\bar{a})$:

```
    b̄ = x^n+1
    q = QUO(b̄, ā)
    if REM(b̄, ā) == 0:
        return 0
    else
        u0 = b̄
        u1 = ā
        v0 = 1
        v1 = q
        while deg(u1) > 0:
            q = QUO(u0, u1)
            u2 = REM(u0, u1)
            v2 = v0 + q * v1
            u0 = u1
            u1 = u2
            v0 = v1
            v1 = v2
        return v1
```