***ASSIGNMENT-2***

***QUESTION NO.:-1***

***ALGORITHM:-***

*CALCULATE (c,m)*

Statement: - In this function take two strings as arguments one is ‘c’ for base and another is ‘m’ for exponent.

Step 1: i←calculate the length of string ‘c’.

Step 2: j←calculate the length of string ‘m’.

Step 3: y←put the starting index value of loop which is 0.

Step 4: Repeat step 5 to 6 while (y<i) do,

Step 5: a[y]←c[y]-‘0’ *//subtract the 0 ASCII value from c[y] that store a character of base*

Step 6: y←y+1

Step 7: y←put the starting index value of loop which is 0.

Step 8: Repeat step 9 to 10 while (y<j) do,

Step 9: k[y]←m[y]-‘0’ *//subtract the 0 ASCII value from m[y] that store a character of exponent*

Step 10: y←y+1

Step 11: t←CHECK(k,j-1)  *//call the CHECK function*

Step 12: If (t=1) then go to step 13 to 16,

Otherwise go to step 17

Step 13: y←put the starting index value of loop which is 0.

Step 14: Repeat step 15 to 16 while (y<i) do,

Step 15: n[y]←c[y]-‘0’ *//subtract the 0 ASCII value from c[y] that store a character of base*

Step 16: y←y+1

Step 17: Print “the value is..:0” and return to the main program.

Step 18: s←i-1 *//here ‘i’ store the size of string ‘c’*

Step 19: t←CHECK(k,j-1) *//call the CHECK function*

Step 20: Repeat step 21 to 58 while (t=1) do,

Step 21: v←initialize to 0

Step 22: y←put the value of ‘i-1’

Step 23: Repeat step 24 to 52 while (y≥0) do,

Step 24: d←initialize by 0

Step 25: q←initialize by 0

Step 26: If (y≠i-1) then go to step 27 to 37,

Otherwise go to step 38 to 46

Step 27: z←initialize by value of ‘s’

Step 28: Repeat step 29 to 36 while (z ≥0) do,

Step 29: u←a[y]\*(n[z])

Step 30: If (v+q≤x) then go to step 31,

Otherwise go to step 32

Step 31: u←u+d+b[v+q]

Step 32: u←u+d

Step 33: d←u/10

Step 34: b[v+q]←u%10

Step 35: q←q+1

Step 36: z←z-1

Step 37: x←v+q-1

Step 38: z←initialize by value of ‘s’

Step 39: Repeat step 40 to 45 while (z ≥0) do,

Step 40: u←a[y]\*(n[z])

Step 41: u←u+d

Step 42: d←u/10

Step 43: b[v+q]←u%10

Step 44: q←q+1

Step 45: z←z-1

Step 46: x←v+q-1

Step 47: If (d>0) then go to step 48 to 49,

Otherwise go to step 50

Step 48: b[v+q]←d

Step 49: n1←v+q

Step 50: n1←v+q-1

Step 51: v←v+1

Step 52: y←y-1

Step 53: y←initialize by 0

Step 54: Repeat step 55 to 56 while (y ≤n1) do,

Step 55: n[n1-y]←b[y]

Step 56: y←y+1

Step 57: s ←nitialize by value of ‘n1’

Step 58: t←CHECK(k,j-1) *//call the CHECK function*

Step 59: show(n,n1) *//call the SHOW function*

Step 60: END

*CHECK (m,d)*

Statement: This function take one string ‘m’ and an integer ‘d’ as arguments which are the string of exponent and length of this string. This function return an integer value.

Step 1: i ←nitialize by value of ‘d’

Step 2: Repeat step 3 to 5 while (i≥0) do,

Step 3: If (m[i]≠0) then go to step 4,

Otherwise continue this loop

Step 4: y← nitialize by value of ‘i’ and go to step 6

Step 5: i←i-1

Step 6: If (y≥0) then go to step 7 to 11,

Otherwise return 0 to the calling function

Step 7: i ←nitialize by value of ‘y+1’

Step 8: Repeat step 9 to 10 while (i≤d) do,

Step 9: m[i]←initialize by 9

Step 10: i←i+1

Step 11: Return 1 to the calling function.

Step 12: END

*SHOW (n,s)*

Statement: This function take an integer array ‘n’ and size of this array ‘s’ as arguments.

Step 1: i←initialize by 0

Step 2: Repeat step 3 to 4 while (i≤s) do,

Step 3: Display the value of ‘n[i]’.

Step 4: i←i+1

Step 5: END

*Main ()*

Step 1: Input the string of base in ‘a’.

Step 2: Input the string of exponent in ‘m’.

Step 3: CALCULATE (a,m) *// call the CALCULATE function*

Step 4: END