***ASSIGNMENT-5***

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

***ALGORITHM:-***

*DISTANCE (a,b,c,d)*

Statement: This function take four inputs as argument to calculate distance.

Step 1: Calculate the absolute value of (a-c) and (b-d) by using ‘abs’ library function.

Step 2: Return the absolute value of ((a-c) + (b-d))

Step 3: END

*Main ()*

Step 1: Initialize ‘r’ by 0.

Step 2: Input the dimensions of the city grid in ‘dx’ and ‘dy’.

Step 3: Input the number of coffee shops in ‘n’ and number of queries in ‘q’.

Step 4: Initialize ‘j’ by 0

Step 5: Repeat step 6 to 7 while (j<n) do,

Step 6: Input the co-ordinate of coffee shop in ‘c[0][j]’ and ‘c[1][j]’.

Step 7: j←j+1

Step 8: Initialize ‘j’ by 0

Step 9: Repeat step 10 to 11 while (j<q) do,

Step 10: Input the query value in ‘qu[j]’.

Step 11: j←j+1

Step 12: Initialize ‘j’ by 0

Step 13: Repeat step 14 to 39 while (j<q) do,

Step 14: Initialize ‘cv’ by 0

Step 15: w←put the value of ‘qu[j]’

Step 16: Initialize ‘k’ by 1

Step 17: Repeat step 18 to 37 while (k<dx) do,

Step 18: Initialize ‘l’ by 1

Step 19: Repeat step 20 to 36 while (l<dy) do,

Step 20: Initialize ‘ccv’ by 0

Step 21: Initialize ‘m’ by 0

Step 22: Repeat step 23 to 24 while (m<n) do,

Step 23: If (DISTENCE (k,l,c[0][m],c[1][m])≤w) then do, *//call the DISTANCE function*

ccv←ccv+1

Step 24: m←m+1

Step 25: If (ccv>cv) then go to step 26 to 28,

Otherwise go to step 29

Step 26: cv←put the value of ‘ccv’

Step 27: cx←put the value of ‘k’

Step 28: cy←put the value of ‘l’

Step 29: If (ccv=cv and l<cy) then go to step 30 to 32,

Otherwise go to step 33 to 35

Steo 30: cv←put the value of ‘ccv’

Step 31: cx←put the value of ‘k’

Step 32: cy←put the value of ‘l’

Step 33: cv←put the value of ‘ccv’

Step 34: cx←put the value of ‘k’

Step 35: cy←put the value of ‘l’

Step 36: l←l+1

Step 37:k←k+1

Step 38: Display the maximum reached coffee shops ‘cv’ and co-ordinate ‘cx’, ‘cy’.

Step 39: j←j+1

Step 40: END

***QUESTION NO.:-2***

***ALGORITHM:-***

//In this program we use a structure it define bellow

struct machine

{

long int D,P,R,G;

}; //

*SORT (n,m)*

Statement: This function take a structure array ‘m’ and number of machine ‘n’ as arguments.

Step 1: Initialize ‘i’ by 0

Step 2: Repeat step 3 to 10 while (i<n-1) do,

Step 3: Initialize ‘j’ by 0

Step 4: Repeat step 5 to 9 while (j<n) do,

Step 5: If (m[j].D<m[i].D) then go to step 6 to 8

Step 6: temp←m[i]

Step 7: m[i]←m[j]

Step 8: m[j]←temp

Step 9: j←j+1

Step 10: i←i+1

Step 11: END

*CALCULATE (m,n,c,d)*

Statement: This function take some arguments which are a structure array ‘m’ and three integers ‘n’, ‘c’, ‘d’ that takes for number of machine, restructuring cost and restructuring day.

Step 1: Initialize ‘i’ and ‘f’ by 0

Step 2: Repeat step 3 to 30 while (i<n-1) do,

Step 3: b←put the value of ‘m[i].D’

Step 4: e←put the value of ‘i’

Step 5: a←put the value of ‘c’

Step 6: If (m[i].P≤c) then go to step 7 to 26,

Otherwise go to step 27

Step 7: a←a-m[i].P

Step 8: j←i+1

Step 9: Initialize ‘k’ by b+1

Step 10: Repeat step 11 to 26 while (k≤d) do,

Step 11: If (j==n) then go to step 28

Step 12: If (k==m[j].D) then go to step 13 to 24,

Otherwise go to step 25

Step 13: If (m[j].P≤a and m[j].G≥m[e].G) then go to step 14 to 22,

Otherwise go to step 23

Step 14: If (m[j].G>m[e].G) then go to step 15 to 17,

Otherwise go to step 18

Step 15: a←a-m[j].P

Step 16: a←a+m[e].R

Step 17: e←put the value of ‘j’

Step 18: If (m[j].G=m[e].G and m[j]>m[e].R) then go to step 19 to 21,

Otherwise go to step 22

Step 19: a←a-m[j].P

Step 20: a←a+m[e].R

Step 21: e←put the value of ‘j’

Step 22: a←a+m[e].G

Step 23: a←a+m[e].G

Step 24: j←j+1

Step 25: a←a+m[e].G

Step 26: k←k+1

Step 27: Go to step 2 with i←i+1

Step 28: arr[f]←a+m[e].R+((d-k+1)\*m[i].G)

Step 29: f←f+1

Step 30: i←i+1

Step 31: If (m[i].P≤c) then go to step 32

Step 32: arr[f]←(c-m[i].P)+(d-m[i].D)\*m[i].G+m[i].R

Step 33: j←put the value of ‘arr[f]’

Step 34: Initialize ‘i’ by ‘f-1’

Step 35: Repeat step 36 to 38 while (i≥0) do,

Step 36: If (j<arr[i]) then go to step 37

Step 37: j←put the value of ‘arr[i]’

Step 38: i←i-1

Step 39: Display the value maximum profit ‘j’

Step 40: END

*MAIN ()*

Step 1: Input the number of machine, restructuring cost and restructuring day in ‘n’, ‘c’, ‘d’ respectively.

Step 2: Initialize ‘i’ by 0

Step 3: Repeat step 4 to 5 while (i<n) do,

Step 4: Input the selling days, brought price, sell price and profit of machine in ‘D’, ‘P’, ‘R’, ‘G’ respectively.

Step 5: i←i+1

Step 6: SORT (n,m) //call SORT function

Step 7: CALCULATE (m,n,c,d) //call CALCULATE function

Step 8: END

Question no: 3

**Algorithm:**

1. Start
2. Define pi=3.142857142

//In function Main()

1. Set area←0,pari←0, a[20] as floating point variable.
2. Set I,n as integer variable
3. Display “Enter number of segments:”
4. Input n
5. Display “Enter length of each segment”
6. Set i←0
7. Repeat steps 10-12 if (i<n)
8. Input a[i]
9. Set pari←a[i]+pari
10. i←i+1
11. Set i=0
12. Repeat steps 15-16 if (i<n)
13. Set area←area+((a[i]/2)\*(a[i]/2)\*(1/tan(pi\*a[i]/pari)))
14. Set i=i+1
15. Display “ Max area that can be covered by given segments:”
16. Display value of ‘area’
17. End