Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.

#include<stdio.h>

int binsrc(int x[],int low,int high,int key)

{

int mid;

while(low<=high)

{

mid=(low+high)/2;

if(x[mid]==key)

return mid;

if(x[mid]<key)

low=mid+1;

else

high=mid-1;

}

return -1;

}

int main()

{

int a[20],key,i,n,succ;

printf("Enter the n value");

scanf("%d",&n);

if(n>0)

{

printf("enter the elements in ascending order\n");

for(i=0;i<n;i++)

scanf("%d",&a[i]);

printf("enter the key element to be searched\n");

scanf("%d",&key);

succ=binsrc(a,0,n-1,key);

if(succ>=0)

printf("Element found in position = %d\n",succ+1);

else

printf("Element not found \n");

}

else

printf("Number of element should be greater than zero\n");

return 0;

}

Design, develop, code and run the program in any suitable language to implement the quick sort algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.

#include<stdio.h>

void quicksort(int x[10],int first,int last)

{

int temp,pivot,i,j;

if(first<last)

{

pivot=first;

i=first;

j=last;

while(i<j)

{

while(x[i]<=x[pivot] && i<last)

i++;

while(x[j]>x[pivot])

j--;

if(i<j)

{

temp=x[i];

x[i]=x[j];

x[j]=temp;

}

}

temp=x[pivot];

x[pivot]=x[j];

x[j]=temp;

quicksort(x,first,j-1);

quicksort(x,j+1,last);

}

}

// main program

int main()

{

int a[20],i,key,n;

printf("enter the size of the array");

scanf("%d",&n);

if(n>0)

{

printf("enter the elements of the array");

for(i=0;i<n;i++)

scanf("%d",&a[i]);

quicksort(a,0,n-1);

printf("the elements in the sorted array is:\n");

for(i=0;i<n;i++)

printf("%d\t",a[i]);

}

else

{

printf(“size of array is invalid\n”);

}

Design, develop, code and run the program in any suitable language to implement an absolute letter grading procedure, making suitable assumptions. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.

#include<stdio.h>

int main()

{

float per;

char grade;

scanf("%f",&per);

if(per>=90)

grade= 'A';

else if(per>=80 && per<90)

grade ='B';

else if(per>=70 && per<80)

grade ='C';

else if(per>=60 && per<70)

grade='D';

else grade='E';

switch(grade)

{

case 'A': printf("\nEXCELLENT"); break;

case 'B':printf("\nVery Good"); break;

case 'C' : printf("\nGood"); break;

case 'D': printf("\nAbove Average"); break;

case 'E': printf("\n Satisfactory"); break;

}

printf("\t The percentage = %f and grade is %c ",per,grade);

return 0;

}

Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive different test cases, execute these test cases and discuss the test results.

2 #include<stdio.h>

3 int main()

4 {

5 int locks, stocks, barrels, tlocks, tstocks, tbarrels;

6 float lprice,sprice,bprice,lsales,ssales,bsales,sales,comm;

7 lprice=45.0;

8 sprice=30.0;

9 bprice=25.0;

10 tlocks=0;

11 tstocks=0;

12 tbarrels=0;

13 printf("\nenter the number of locks and to exit the loop enter -1 for locks\n");

scanf("%d", &locks);

14 while(locks!=-1) {

15 printf("enter the number of stocks and barrels\n");

scanf("%d%d",&stocks,&barrels);

16 tlocks=tlocks+locks;

17 tstocks=tstocks+stocks;

18 tbarrels=btarrels+barrels;

19 printf("\nenter the number of locks and to exit the loop enter -1 for

locks\n"); scanf("%d",&locks);

20 }

21 printf("\ntotal locks = %d\”,tlocks);

22 printf(“total stocks =%d\n”,tstocks);

23 printf(“total barrels =%d\n",tbarrels);

24 lsales = lprice\*tlocks;

25 ssales=sprice\*tstocks;

26 bsales=bprice\*tbarrels;

27 sales=lsales+ssales+bsales;

28 printf("\nthe total sales=%f\n",sales);

29 if(sales > 1800.0)

30 {

31 comm=0.10\*1000.0;

32 comm=comm+0.15\*800;

33 comm=comm+0.20\*(sales-1800.0);

}

34 else if(sales > 1000)

35 {

36 comm =0.10\*1000;

37 comm=comm+0.15\*(sales-1000);

}

38 else

39 comm=0.10\*sales;

40 printf("the commission is=%f\n",comm);

41 return 0;

42 }