|  |  |  |
| --- | --- | --- |
| **Name:**  **P K Navin Shrinivas** | **SRN:PES2UG20CS237** | **Section:G-2(B-2)** |
| **Date:8/07/21** | **Week Number:10** |

|  |  |
| --- | --- |
| **1** | Implement Binary Search using call back when there is more than one constraint to check for.  a) Search for a number if the number is even  b) Search for a number if the number is less than 22.  **Input:**  enter the element to be searched  18  **Output:**  It is even and found at 2 position  It is less than 22 and found at 2 position  **Input:**  enter the element to be searched  56  **Output:**  It is even and found at 8 position  not found  **Input:**  enter the element to be searched  53  **Output:**  not found  not found |
|  | **Program:**  #include<stdio.h>  *//a-array of numbers*  *//n-number to be searched*  *//l->r is the range in which element is to be searched*  int searchcall(int (\*fn)(int\* a,int n,int l,int r),int\* *a*,int *n*,int *l*,int *r*)  {  return fn(a,n,l,r);  }  int evensearch(int\* *a*,int *n*,int *l*,int *r*)  {  if(n%2!=0)  return -1;  if(l<=r)  {  int center=(r+l)/2;  if(\*(a+center)==n)  return center;  else if(\*(a+center)>n)  evensearch(a,n,l,center);  else  evensearch(a,n,center+1,r);  }  else  return -1;  }  int less22search(int\* *a*,int *n*,int *l*,int *r*)  {  if(n>22)  return -1;  else if(l<=r)  {  int center=(r+l)/2;  if(\*(a+center)==n)  return center;  else if(\*(a+center)>n)  less22search(a,n,l,center);  else  less22search(a,n,center+1,r);  }  else  return -1;  }  int main()  {  *//NOTE : I implement callback without function pointer unlike what she taught in class*  int n;  printf("Enter number of elements : ");  scanf("%d",&n);int a[n];  printf("Enter the elements :\n");  for(int i=0;i<n;i++)  {  scanf("%d",a+i);  }  int searchnum;  printf("Element to be searched : ");  scanf("%d",&searchnum);  if(searchcall(evensearch,a,searchnum,0,n-1)>=0)  {  printf("It is even and found at position %d. \n",searchcall(evensearch,a,searchnum,0,n-1)+1);  }  else{  printf("Not found!\n");  }  if(searchcall(less22search,a,searchnum,0,n-1)>=0)  {  printf("It is less than 22 and found at position %d.",searchcall(less22search,a,searchnum,0,n-1)+1);  }  else{  printf("Not found!");  }  } |
|  | **Output Screenshot:**  **1** |
| 2 | Write a program to copy the contents of one file to another using command line arguments  (Instruction to be given in the command line)  >a abc.txt def.txt  (abc.txt is the file having contents which will be copied to the file def.txt) |
|  | **Program:**  #include<stdio.h>  int main(int *argc*, char\* *argv*[])  {  if(argc<=2)  {  printf("Not sufficient arguments!Enter names of two file");  return 0;  }  FILE\* f1=fopen(argv[1],"r"); *//0 index is the command itself*  FILE\* f2=fopen(argv[2],"w+");  char contents[10000];  fscanf(f1,"%[^NULL]%\*c",contents);  fprintf(f2,"%s",contents);  printf("Contents copied from %s to %s",argv[1],argv[2]);  } |
|  | **Output Screenshots:**  **2**  **2file1.txt:**  These are the contents of file 1.  **2file2.txt [After executing program]:**  These are the contents of file 1. |
| 3 | Write a program using enumerated types which when given today's date will print out tomorrow's date.  **Input:**  Enter a date (number 3 letter lower case month e.g. 31 jan)  30 nov  **Output:**  Tomorrow is 1 dec  **Input:**  Enter a date (number 3 letter lower case month e.g. 31 jan)  31 dec  **Output:**  Tomorrow is 1 jan |
|  | **Program:**  #include<stdio.h>  #include<string.h>  enum month{rand,jan,feb,mar,apr,may,jun,jul,aug,sep  ,oct,nov,dec};  struct enummap{  char\* usermon;  enum month enummon;  }maptable[]={{"jan",jan},{"feb",feb},{"mar",mar},  {"apr",apr},{"may",may},{"jun",jun},  {"jul",jul},{"aug",aug},{"sep",sep},  {"oct",oct},{"nov",nov},{"dec",dec}};  int nextflag;  void next(char *day*,enum month *mon*)  {  if(mon==2 && day==29) *//we dont have year so assuming years are never leap*  nextflag=1;  if(mon<8)  if(mon%2==0 && day==30)  nextflag=1;  if(mon%2!=0 && day==31)  nextflag=1;  if(mon>=8)  if(mon%2!=0 && day==30)  nextflag=1;  if(mon%2==0 && day==31)  nextflag=1;  }  int main()  {  int day;  enum month m;  char mon[3];  printf("Enter date in the format DD MMM [e.g. 31 jan] :");  scanf("%d %[^\n]%\*c",&day,mon);  for(int i=0;i<12;i++)  {  if(strcmp((maptable+i)->usermon,mon)==0)  m=(maptable+i)->enummon;  }  next(day,m);  if(nextflag) *//assuming user always gives a valid date*  {  m+=1;  if(m==13)*//demn i did not think of this case , that example in the file is sneaky!*  m=1;  char\* nextmon;  for(int i=0;i<12;i++)  {  if((maptable+i)->enummon-m==0)  nextmon=(maptable+i)->usermon;  }  printf("Next day is : %d %s",1,nextmon);  }  else{  printf("Next day is : %d %s",day+1,mon);  }  } |
|  | **Output Screenshot:**  **3** |