System Programming

PRACTICAL FILE

(CSX-326)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Dr. B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY

JALANDHAR – 144011, PUNJAB (INDIA)

January - May 2017

Submitted To:

Ms Kamalpreet Kaur
Assistant Professor
Dept. Of CSE

Submitted By:

Sai ganesh munduru
14103021
3RD year CSE

LAB-1 17-Jan-2017

Aim:- *To implement linear search in array.*

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   int n;
   cout<<"Enter the number of elements in the array\n";
   cin>>n;
   int ar[n];
   cout<<"Now enter the elements\n";
   for(int i=0;i<n;i++)
      cin>>ar[i];
   cout<<"Enter the element to be searched\n";
   int c;
   cin>>c;
   int flag=0;
   for(int i=0;i<n;i++)
   {
      if(ar[i]==c)
      {
         flag=1;
         cout<<"Element found at index "<<i+1<<"\n";
      }
   }
   if(flag==0)
      cout<<"No element found\n"; }
```

```
Enter the number of elements in the array
10
Now enter the elements
34 45 12 97 35 43 1212 677 342 100
Enter the element to be searched
12
Element found at index 3
```

Aim:- To implement binary search in an array.

```
#include<bits/stdc++.h>
using namespace std;
int binary(int ar[],int l,int h,int c)
{
   int mid=l+(h-l)/2;
   if(ar[mid]==c)
   {
      cout<<"Element found!\n";
      return 0;
   }
   if(l==h)
      return -1;
   if(ar[mid]>c)
      binary(ar,l,mid,c);
   else
      binary(ar,mid+1,h,c);
}
int main()
{
```

```
    int n;
    cout<<"Enter the number of elements in the array\n";
    cin>>n;
    int ar[n];
    cout<<"Now enter the elements\n";
    for(int i=0;i<n;i++)
        cin>>ar[i];
    cout<<"Enter the element to be searched\n";
    int c;
    cin>>c;
    sort(ar,ar+n);
    if(binary(ar,0,n-1,c)==-1)
        cout<<"No element found\n";
}
```

```
Enter the number of elements in the array
10
Now enter the elements
34 45 12 97 35 43 1212 677 342 100
Enter the element to be searched
12
Element found!
```

Aim:- To implement bubble sort.

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   int ar[100];
   int n,temp;
   cout<<"Enter the number of elements in the array\n";
   cin>>n;
   cout<<"Now enter the elements\n";
   for(int i=0; i<n; i++)
      cin>>ar[i];
   for(int i=0; i<n; i++)
   {
      for(int j=0; j<n-i; j++)
      {
         if(ar[j]>ar[j+1])
         {
            temp=ar[j+1];
            ar[j+1]=ar[j];
            ar[j]=temp;
         }
      }
   }
   cout<<"The sorted array is\n";
   for(int i=0; i<n; i++)
      cout<<ar[i]<<" ";
```

```
}
```

```
Enter the number of elements in the array
10
Now enter the elements
12 23 1 323 21 9023 78 3233 56 32
The sorted array is
1 12 21 23 32 56 78 323 3233 9023
```

Aim:- To implement selection sort.

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   int ar[100];
   int n,temp,num,k;
   cout<<"Enter the number of elements in the array\n";
   cin>>n;
   cout<<"Now enter the elements\n";
   for(int i=0; i<n; i++)
      cin>>ar[i];
   for(int i=0; i<n; i++)
   {
      int minn=INT_MAX;
      for(int j=i; j<n; j++)
      {
         if(ar[j]<minn)
         {
            minn=ar[j];
            k=j;
         }
```

```
        }
        temp=ar[i];
        ar[k]=temp;
        ar[i]=minn;
    }
    cout<<"The sorted array is\n";
    for(int i=0; i<n; i++)
        cout<<ar[i]<<" ";
    cout<<endl;
}
```

```
Enter the number of elements in the array
10
Now enter the elements
12 23 23432 64 324 67 2313 98 100 34
The sorted array is
12 23 34 64 67 98 100 324 2313 23432
```

Aim:- To implement insertion sort.

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int ar[100];
    int n;
    cout<<"Enter the number of elements in the array\n";
    cin>>n;
    cout<<"Now enter the elements\n";
    for(int i=0; i<n; i++)
        cin>>ar[i];
```

```
    for(int i=1; i<n; i++)
    {
        int j=i-1;
        int key=ar[i];
        while(j>=0&&ar[j]>key)
        {
            ar[j+1]=ar[j];
            j--;
        }
        ar[j+1]=key;
    }
    cout<<"The sorted array is\n";
    for(int i=0; i<n; i++)
        cout<<ar[i]<<" ";
}
```

```
Enter the number of elements in the array
10
Now enter the elements
12 23 23432 64 324 67 2313 98 100 34
The sorted array is
12 23 34 64 67 98 100 324 2313 23432
```

Aim:- To implement heap sort.

```
#include<bits/stdc++.h>
using namespace std;
int ar[1000010];
void swap(int *a,int *b)
{
    int temp=*a;
```

```
    *a=*b;
    *b=temp;
}
void heapify(int ar[],int l,int n)
{
    int k=2*l;
    if(k+1<=n)
    {
        if((ar[k]>ar[l])&&(ar[k]>ar[k+1]))
        {
            swap(&ar[k],&ar[l]);
            heapify(ar,k,n);
        }
        else if((ar[k+1]>ar[l])&&(ar[k+1]>=ar[k]))
        {
            swap(&ar[k+1],&ar[l]);
            heapify(ar,k+1,n);
        }
    }
    else if(k<=n&&ar[k]>ar[l])
        swap(&ar[k],&ar[l]);
}
void heap(int ar[],int n)
{
    for(int i=n/2; i>=1; i--)
        heapify(ar,i,n);
}
```

```
int main()
{
   int n;
   cout<<"Enter the number of elements in the array\n";
   cin>>n;
   cout<<"Now enter the elements\n";
   for(int i=1; i<=n; i++)
      scanf("%d",&ar[i]);
   heap(ar,n);
   swap(&ar[n],&ar[1]);
   for(int i=n-1; i>=1; i--)
   {
      heapify(ar,1,i);
      swap(&ar[1],&ar[i]);
   }
   cout<<"The sorted array is\n";
   for(int i=1; i<=n; i++)
      printf("%d\n",ar[i]);
}
```

```
Enter the number of elements in the array
10
Now enter the elements
12 23 23432 64 324 67 2313 98 100 34
The sorted array is
12 23 34 64 67 98 100 324 2313 23432
```

Aim:- TO IMPLEMENT PASS 1

```
#include<stdio.h>
#include<stdlib.h>
 #include<curses.h>
#include<string.h>
int main()
{
  FILE *f1,*f2,*f3,*f4,*f5;
  int location_counter,starting_address,i=0,j=0,loc_coun_arr[10],program_length,len,k,l=0;

  char name[10],operand[10],variables[10],instruction[10],s1[10],instruction1[10],operand1[10];
  char location_counter_string[10],ms[10];
  char symbols[10],symbol_address[10],temp_1[10],temp_2[10],s2[10],q[10],s3[10];

 f1=fopen("input.txt","r");
 f2=fopen("optab.txt","r");

 f3=fopen("symtab.txt","w+");
 f4=fopen("symtab1.txt","w+");
 f5=fopen("output.txt","w+");
 fscanf(f1,"%s%s%s",variables,instruction,operand);
 if(strcmp(instruction,"START")==0)
{
   starting_address=atoi(operand);
   strcpy(name,variables);
   location_counter=starting_address;
}
strcpy(s1,"*");
```

```
fscanf(f1,"%s%s%s",variables,instruction,operand);
while(strcmp(instruction,"END")!=0)
{
 if(strcmp(variables,"-")==0)
 {
  fscanf(f2,"%s%s",instruction1,operand1);
  while(!feof(f2))
  {
   if(strcmp(instruction1,instruction)==0)
   {
  loc_coun_arr[i]=location_counter+1;
  fprintf(f3,"%s\t%s\n",operand,s1);
  fprintf(f5,"%s\t0000\n",operand1);
  location_counter=location_counter+3;
  i=i+1;
  break;
   }
   else
  fscanf(f2,"%s%s",instruction1,operand1);
  }
 }
 else
 {
  fseek(f3,SEEK_SET,0);
  fscanf(f3,"%s%s",symbols,symbol_address);
  while(!feof(f3))
  {
   if(strcmp(symbols,variables)==0)
```

```
    {

  sprintf(location_counter_string, "%d", location_counter);
  fprintf(f4,"%s\t%s\n",variables,location_counter_string);
  sprintf(ms, "%d", loc_coun_arr[j]);
  j++;
  fprintf(f5,"%s\t%s\n",ms,location_counter_string);
  i++;
  break;
    }
    else
  fscanf(f3,"%s%s",symbols,symbol_address);
  }
  if(strcmp(instruction,"RESW")==0)
   location_counter=location_counter+3*atoi(operand);
  else if(strcmp(instruction,"BYTE")==0)
  {
   strcpy(s2,"-");
   len=strlen(operand);
   location_counter=location_counter+len-2;
   for(k=2;k<len;k++)
   {
   q[l]=operand[k];
   l=l+1;
   }
   fprintf(f5,"%s\t%s\n",q,s2);
   break;
  }
```

```
    else if(strcmp(instruction,"RESB")==0)
     location_counter=location_counter+atoi(operand);
    else if(strcmp(instruction,"WORD")==0)
    {
      strcpy(s3,"#");
      location_counter=location_counter+3;
      fprintf(f5,"%s\t%s\n",operand,s3);
      break;
    }
   }
    fseek(f2,SEEK_SET,0);
    fscanf(f1,"%s%s%s",variables,instruction,operand);
 }
 fseek(f5,SEEK_SET,0);
 program_length=location_counter-starting_address;
 fscanf(f5,"%s%s",temp_1,temp_2);
 fseek(f5,SEEK_SET,0);
 fscanf(f5,"%s%s",temp_1,temp_2);
getch();
return 0;
}
```

NPUT.txt OPTAB.txt

```
COPY    START   1000
-       LDA     ALPHA
-       STA     BETA
ALPHA   RESW    1
BETA    RESW    1
-       END     -
```

```
LDA     00
STA     23
LDCH    15
STCH    18
```

Aim:- TO IMPLEMENT PASS 2

```
 #include<stdio.h>
 #include<curses.h>
#include<string.h>
#include<stdlib.h>
int main()
{
  char a[10],ad[10],label[10],opcode[10],operand[10],symbol[10],ch;  int
st,diff,i,address,add,len,actual_len,finaddr,prevaddr,j=0,s;
  char mnemonic[15][15]={"LDA","STA","LDCH","STCH"};
  char code[15][15]={"33","44","53","57"};
  FILE *fp1,*fp2,*fp3,*fp4;

  fp1=fopen("ASSMLIST.DAT","w");
  fp2=fopen("SYMTAB.DAT","r");
  fp3=fopen("INTERMED.DAT","r");
  fp4=fopen("OBJCODE.DAT","w");
  fscanf(fp3,"%s%s%s",label,opcode,operand);

  while(strcmp(opcode,"END")!=0)
  {
   prevaddr=address;
   fscanf(fp3,"%d%s%s%s",&address,label,opcode,operand);
  }
  finaddr=address;
  fclose(fp3);
  fp3=fopen("INTERMED.DAT","r");

  fscanf(fp3,"%s%s%s",label,opcode,operand);
  if(strcmp(opcode,"START")==0)
  {
   fprintf(fp1,"\t%s\t%s\t%s\n",label,opcode,operand);
   fprintf(fp4,"H^%s^00%s^00%d\n",label,operand,finaddr);
   fscanf(fp3,"%d%s%s%s",&address,label,opcode,operand);
   st=address;
   diff=prevaddr-st;
   fprintf(fp4,"T^00%d^%d",address,diff);
  }
  while(strcmp(opcode,"END")!=0)
  {
   if(strcmp(opcode,"BYTE")==0)
```

```
{
 fprintf(fp1,"%d\t%s\t%s\t%s\t",address,label,opcode,operand);
 len=strlen(operand);
 actual_len=len-3;
 fprintf(fp4,"^");
 for(i=2;i<(actual_len+2);i++)
 {
 // itoa(operand[i],ad,16);
   sprintf(ad, "%d", operand[j]);
  fprintf(fp1,"%s",ad);
  fprintf(fp4,"%s",ad);
 }
 fprintf(fp1,"\n");
}
else if(strcmp(opcode,"WORD")==0)
{
 len=strlen(operand);
  s = atoi(operand);
 //itoa(s,a,10);
 sprintf(a, "%d", s);
 fprintf(fp1,"%d\t%s\t%s\t%s\t00000%s\n",address,label,opcode,operand,a);
 fprintf(fp4,"^00000%s",a);
}
else if((strcmp(opcode,"RESB")==0)||(strcmp(opcode,"RESW")==0))
 fprintf(fp1,"%d\t%s\t%s\t%s\n",address,label,opcode,operand);
else
{
 while(strcmp(opcode,mnemonic[j])!=0)
  j++;
 if(strcmp(operand,"COPY")==0)
  fprintf(fp1,"%d\t%s\t%s\t%s\t%s0000\n",address,label,opcode,operand,code[j]);
 else
 {
  rewind(fp2);
  fscanf(fp2,"%s%d",symbol,&add);
   while(strcmp(operand,symbol)!=0)
    fscanf(fp2,"%s%d",symbol,&add);
  fprintf(fp1,"%d\t%s\t%s\t%s\t%s%d\n",address,label,opcode,operand,code[j],add);
  fprintf(fp4,"^%s%d",code[j],add);
 }
}
```

```
 fscanf(fp3,"%d%s%s%s",&address,label,opcode,operand);
 }
 fprintf(fp1,"%d\t%s\t%s\t%s\n",address,label,opcode,operand);
 fprintf(fp4,"\nE^00%d",st);
 printf("\n Intermediate file is converted into object code");
 printf("\n\nThe contents of Intermediate file:\n\n\t");
 fp3=fopen("INTERMED.DAT","r");
 ch=fgetc(fp3);
 while(ch!=EOF)
 {
  printf("%c",ch);
  ch=fgetc(fp3);
 }
 printf("\n\nThe contents of Symbol Table :\n\n");
 fp2=fopen("SYMTAB.DAT","r");
 ch=fgetc(fp2);
 while(ch!=EOF)
 {
  printf("%c",ch);
  ch=fgetc(fp2);
 }
 printf("\n\nThe contents of Output file :\n\n");
 fp1=fopen("ASSMLIST.DAT","r");
 ch=fgetc(fp1);
 while(ch!=EOF)
 {
  printf("%c",ch);
  ch=fgetc(fp1);
 }
 printf("\n\nThe contents of Object code file :\n\n");
 fp4=fopen("OBJCODE.DAT","r");
 ch=fgetc(fp4);
 while(ch!=EOF)
 {
  printf("%c",ch);
  ch=fgetc(fp4);
 }
 fcloseall();
 getch();
}
```

INTERMEDIATE.DAT

```
ankit@don: /media/ankit/New
COPY    START   2000
2000    **      LDA     FIVE
2003    **      STA     ALPHA
2006    **      LDCH    CHARZ
2009    **      STCH    C1
2012    ALPHA   RESW    1
2015    FIVE    WORD    5
2018    CHARZ   BYTE    C'EOF'
2019    C1      RESB    1
2020    **      END     **
~
```

SYMTAB.DAT

```
ankit@don: /media/ankit/
ALPHA   2012
FIVE    2015
CHARZ   2018
C1      2019
~
~
~
```

OUTPUT

```
Open
H^COPY^002000^002020
T^002000^19^332015^442012^532018^572019^000005^797979
E^002000
```