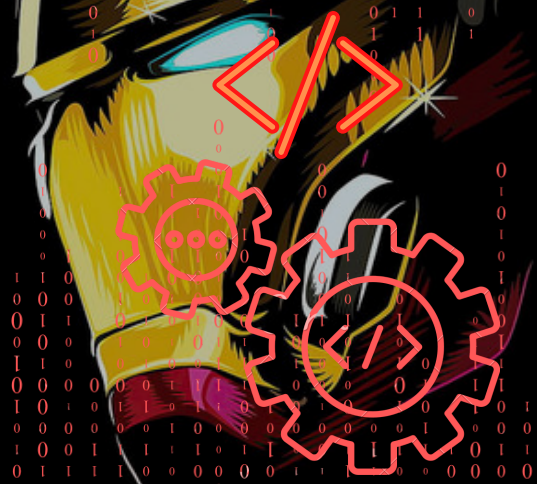
2023

CYBER TRON

C-PROGRAMMING



(THIRD PART)

T.I.M. HAMIM

Index:

1. linear search

Linear search in C to find whether a number is present in an array. If it's present, then at what location it occurs. It is also known as a sequential search. It is straightforward and works as follows: we compare each element with the element to search until we find it or the list ends.

- **Make a program that will take any of integer numbers you want into your database and if you want to find any number in your database it will show you using Linear Search.**

```
//Linear search
#include <stdio.h>
int main()
{
 int value,position,i,n,j;
 printf("Sir, how many numbers do you want to
enter into your database :");
 scanf("%d",&n);
 int a[n];
 printf("Boss, please enter your all numbers into
your database :\n");
 for(i=0;i<n;i++){
 printf("%d no. number =",i+1);
 scanf("%d",&a[i]);
 }
 printf("Please enter your value to find in your
database :");
 scanf("%d",&value);
 for(i=0;i<n;i++){
 if(value==a[i]){
  position=i+1;
  printf("The value is found at %d no.\n",position);
 }
 }
 if(position==-1){
 printf("The number is not found.\n");
 }
 return 0;
}
```

- **Make a program that will find the integer number you want from a unsorted array using linear Search.**

```
#include<stdio.h>
int linearsearch(int arr[], int size, int element){
  // Start searching
   for(int i=0; i< size; i++)
   {
      if(arr[i]==element){
         return i;
      }
   }
   // Searching ends
   return -1;
}

int main(){
 // Unsorted array for linear search
   int arr[]={1,3,5,56,4,3,23,5,4,54634,56,34};
   int size=sizeof(arr)/sizeof(int);
   int element=4;
   int searchIndex=linearsearch(arr,size,element);
   printf("The element %d was found at index %d
\n",element,searchIndex);
   return 0;
}
```

2. binary search

Binary search in C language to find an element in a sorted array. If the array isn't sorted, you must sort it using a sorting technique such as merge sort. If the element to search is present in the list, then we print its location. The program assumes that the input numbers are in ascending order.

- **Make a program that will find the integer number you want from a sorted array using Binary Search.**

```
//binary search
#include <stdio.h>
int main()
{
int ara[] = {1, 4, 6, 8, 9, 11, 14, 15, 20, 25, 33, 83, 87,
97, 99, 100};
int low_indx = 0;
int high_indx = 15;
int mid_indx;
int num = 97;
while (low_indx <= high_indx) {
mid_indx = (low_indx + high_indx) / 2;
if (num == ara[mid_indx]) {
break;
}
if (num < ara[mid_indx]) {
high_indx = mid_indx - 1;
}
else {
low_indx = mid_indx + 1;
}
}
if (low_indx > high_indx) {
printf("%d is not in the array\n", num);
}
else {
printf("%d is found in the array. It is the %d th
element of the array.\n", ara[mid_indx], mid_indx);
}
return 0;
}
```

- **Make a program that will find the integer number you want from a sorted array using Binary Search.**

```
#include<stdio.h>
int binarysearch(int arr[], int size, int element){
   int low, mid, high;
   low=0;
   high=size-1;
   // Start searching
   while(low<=high)
   {
      mid=(low+high)/2;
      if(arr[mid]==element){
         return mid;
      }
      if(arr[mid]<element){
         low=mid+1;
      }
      else{
         high=mid-1;
      }
   }
   // Searching ends
   return -1;
}

int main(){
   // Sorted array for binary search
   int arr[]={1,3,5,56,64,73,123,225,444};
   int size=sizeof(arr)/sizeof(int);
```

```
    int element=56;
    int searchIndex=binarysearch(arr,size,element);
    printf("The element %d was found at index %d
\n",element,searchIndex);
    return 0;
}
```

3. File

In C programming, File is place on disk where a group of related data is stored.

File is a structure which is stored in stdio.h header.

Declaration of file:
FiILE *name_of_file;

In C, you can perform four major operation on the file, either text or binary:

1. Creating a new file,
2. Opening an existing file,
3. Closing a file,
4. Reading from and writing information to a file.

To write something in a file:
fputc(),
fputw(),
fputs(),
fprintf(),
fwrite().

To read something in a file:
fgetc(),
fgetw(),
fgets(),
fscanf(),
fread().

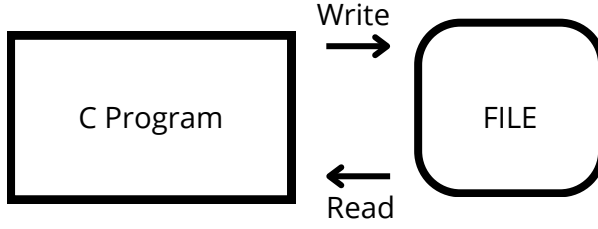
File Opening Models:

Mode	Description
r	Opening an existing text file for reading purpose.
w	Opens a text file for writing. If it does not exist, then a new file is created. Here your program will start writing content from the beginning of the file.
a	Opens a text file for writing in appending mode. If it does not exist , then a new file is created your program will start appending content in the existing file content.
r+	Opens a text file for both reading and writing. If the file exists, loads it loads it into memory and set up a pointer to the first character in it. If file doesn't exist it returns null.
w+	Opens a text file for both reading and writing. If file is present, it first destroys the file to zero length if it exists, otherwise create a file if it does not exist.
a+	Opens a text file for both reading and writing. It creates the file if it does not exist. The reading will start from the beginning but writing can only be appended.

The Random Access Memory is volatle and its content is lost once the program terminates in order to persist the data former we use files.

A file is data stored in a storage device. A C program can talk to file by reading content from it and writing to it.

Programmer



File pointer:

The "FILE" is a structure which needs to be created for opening the file.
A file pointer is a pointer to this structure to the file.

FILE pointer is needed for communication between the file and the program.

A FILE pointer can be created as follows:
FILE *ptr;
ptr = fopen ("file_name.txt" , "mode") ;

FILE opening modes in C:

C offers the programmers to select a mode fo opening a file.

Following modes are primarily used in C FILE I/O :

"r" : Open for reading → If the file does not exist fopen returns NULL

"rb" : Open for reading in binary →

"w" : Open for writing → If the file exist, the contents will be overwrite

"wb" : Open for writing in binary →

"a" : Open for append → If the file does not exist it will be created

Types of Files:

there are two types of files:
1) Text files (.txt , .c)
2) Binary files (.jpg , .dat)

Reading a file:

A file can be opened for reading as follows:

```
FILE *ptr;
ptr = fopen ("Hamim.txt") ;
int num;
```

Let us assume that "Hammim.txt" contains an integer we can read that integer using:
fscanf(ptr , "%D" , &num);

This will read an integer from file in num variable.

Quick Quing: Modify the program above to check whether the file exist or not before opening the file.

Closing the file:

It is very important to close the file after read or write. This is achieved using close as follows:

fclose(ptr) ;

This will tell the compiler that we are done working with this file and the associated resources called be freed.

Write to a file:

We can write to a file in a very similar manner like we read the file.

FILE *fptr ;
fptr = fopen ("Hamim.txt" , "w") ;
int num = 432;

```
fprintf (fptr , "%d" , num) ;
fclose (fptr) ;
```

fgetc() and fputc() :

fgetc() and fputc() are used to read and write a character from/to a file.

```
fgetc(ptr)          =>  Used to read a character from
                        file.
fputc('c' , ptr) ; =>  Used to write character 'c' to
                       the file.
```

EOF (End of File) :

fgetc returns EOF when all the characters from a file have been read. So we can write a check like bellow to detect end of file.

```
while (1) {                          => When all the content
        ch = fgetc (ptr) ;              of a file has been
         if (ch == EOF) {               read, break the loop!
               break ;
         }
     // code
}
```

- **Open a file and then close it.**

```
#include <stdio.h>
int main()
{
   FILE *file;
   file = fopen("1test.txt", "w");
   if (file == NULL)
   {
      printf("File doesn't exist\n");
   }
   else
   {
      printf("File is opened\n");
      fclose(file);
   }
   return 0;
}
```

- **Write your name in a file using fputc() function.**

```
#include <stdio.h>
#include<string.h>
int main()
{
   FILE *file;
   char name[20]="Hamim Talukdaar";
   int str_length=strlen(name),i;
   file = fopen("1test.txt", "w");
   if (file == NULL)
   {
      printf("File doesn't exist\n");
   }
   else
   {
      printf("File is opened\n");
      for(i=0;i<str_length;i++){
        fputc(name[i],file);
      }
      printf("File is writtent successfully.\n");
      fclose(file);
   }
   return 0;
}
```

- **Write your name in a file using fputs() function.**

```
#include <stdio.h>
#include<string.h>
int main()
{
   FILE *file;
   char name[50];
   int str_length=strlen(name),i;
   file = fopen("1test.txt", "w");
   if (file == NULL)
   {
      printf("File doesn't exist\n");
   }
   else
   {
      printf("File is opened\n");
      printf("Enter your name: ");
      gets(name);
      fputs(name,file);
      printf("File is writtent successfully.\n");
      fclose(file);
   }
   return 0;
}
```

- **Make a file program that can take your name and your favorite person name then can keep it in a file using fputs() function.**

```
#include <stdio.h>
#include<string.h>
int main()
{
   FILE *file;
   char name[50];
   int str_length=strlen(name),i;
   file = fopen("1test.txt", "a");
   if (file == NULL)
   {
      printf("File doesn't exist\n");
   }
   else
   {
      printf("File is opened\n");
      printf("Enter your name: ");
      gets(name);
      fputs(name,file);
      fputs("\n",file);
      printf("Enter your favourite person name: ");
      gets(name);
      fputs(name,file);
      fputs("\n",file);
      printf("File is writtent successfully.\n");
      fclose(file);
   }
   return 0;
}
```

- **Write your name in a file using fprintf() function.**

```
#include <stdio.h>
#include<string.h>
int main()
{
   FILE *file;
   char name[50];
   int str_length=strlen(name),i;
   int age;
   file = fopen("1test.txt", "w");
   if (file == NULL)
   {
      printf("File doesn't exist\n");
   }
   else
   {
      printf("File is opened\n");
      printf("Enter your name: ");
      scanf(" %[^\n]",name);
      printf("Enter your age: ");
      scanf("%d",&age);
      fprintf(file,"Name = %s, Age = %d\n",name,age);
      printf("File is writtent successfully.\n");
      fclose(file);
   }
   return 0;
}
```

- **Reading a file using fgetc() function.**

```
#include <stdio.h>
#include<string.h>
int main()
{
   FILE *file;
   char ch;
   file = fopen("1test.txt", "r");
   if (file == NULL)
   {
      printf("File doesn't exist\n");
   }
   else
   {
      printf("File is opened\n");
      while(!feof(file))    // After end of the  file
      {
        ch = fgetc(file);
        printf("%c",ch);
      }
      fclose(file);
   }
   return 0;
}
```

- **Reading a file using fgets() function.**

```
#include <stdio.h>
#include<string.h>
int main()
{
   FILE *file;
   char ch[50];
   file = fopen("1test.txt", "r");
   if (file == NULL)
   {
      printf("File doesn't exist\n");
   }
   else
   {
      printf("File is opened\n");
      while(!feof(file))
      {
      fgets(ch,33, file); // 39 = per line size
      printf("%s\n",ch);
      }
      fclose(file);
   }
   return 0;
}
```

- **Reading a file using fscanf() function.**

```
#include <stdio.h>
#include<string.h>
int main()
{
   FILE *file;
   char ch[50];
   file = fopen("1test.txt", "r");
   if (file == NULL)
   {
      printf("File doesn't exist\n");
   }
   else
   {
      printf("File is opened\n");
      fscanf(file," %[^\n]",&ch);
      printf("%s\n",ch);
      fclose(file);
   }
   return 0;
}
```

- **Reading a file using fscanf() function.**

```
#include <stdio.h>
#include<string.h>
int main()
{
  FILE *file;
  char first_name[50];
  char last_name[50];
  file = fopen("1test.txt", "r");
  if (file == NULL)
  {
    printf("File doesn't exist\n");
  }
  else
  {
    printf("File is opened\n");
    fscanf(file,"%s %s",&first_name,&last_name);
    printf("%s %s\n",first_name,last_name);
    fclose(file);
  }
  return 0;
}
```

- **Open a file and write students name, age and phone number like given bellow**

Name	Age	Phone number
.....	.....	.....
Hamim	22	1731767273
Jim	18	1731767274
Rahim	18	1731767275

```
#include <stdio.h>
#include <string.h>
int main()
{
  FILE *file;
  char name[50];
  int age, phone_Number, num, i;
  file = fopen("student.txt", "a");
  if (file == NULL)
  {
    printf("File doesn't exist\n");
  }
  else
  {
    printf("File is opened\n");
    printf("Enter number of students: ");
    scanf("%d",&num);
    for(i=1;i<=num;i++)
    {
```

```
      printf("Enter student Name : ");
      scanf("%s",&name);
      printf("Enter student Age : ");
      scanf("%d",&age);
      printf("Enter student Phone number : ");
      scanf("%d",&phone_Number);

fprintf(file,"\n%s\t\t%d\t%d\n",name,age,phone_
Number);
    }

    fclose(file);
  }
  return 0;
}
```

- **Write a program to read three integers from a file.**

- **Write a program to generate multiplication table of a given number in text formate. Make sure that the file is readable and well formatted.**

- **Write a program to read a text file character by character and write its content twice in a separate file.**

- **Take name and salary of two employees as input from the user and write them to a text file in the following format:**

  name1, 3300
  name2, 7700

- **Write a program to modify a file containing an integer to double its value.**

2 => 4
prev. file new file

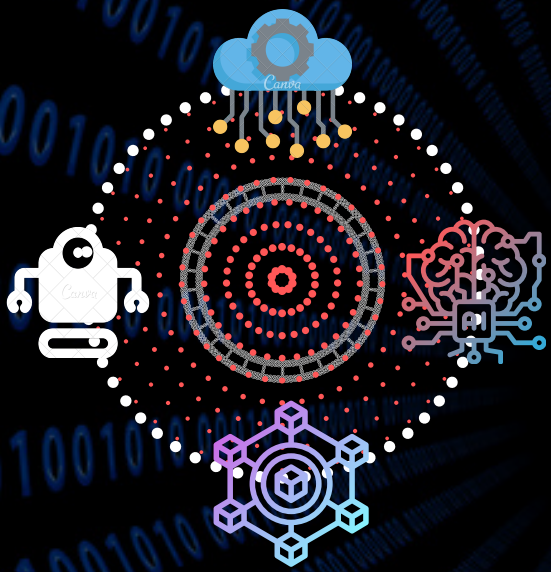
- **Make a program that can open a file.**

```
#include<stdio.h>
int main()
{
   FILE *file;
   file = fopen("test.txt","w");
   if(file==NULL)
      printf("File doesn't exist");
   else
      printf("File is opend");
      fclose(file);
    return 0;
}
```

- **Make a program in which we can write something using file.**

```
#include<stdio.h>
#include<string.h>
int main()
{
   FILE *file;
   char name[20]="Hamim Talukder";
   int length = strlen(name), i;
   file = fopen("test.txt","w");
   if(file==NULL)
   {
      printf("File doesn't exist");
   }
   else
```

```
    {
        printf("File is opend\n");
        for(i=0;i<length;i++){
            fputc(name[i],file);
        }
        printf("File is written successfully.\n");
        fclose(file);
    }
     return 0;
}
```



C-PROGRAMMING
(THIRD PART)
T.I.M. HA-MEAM

ABC
PROKASHONI