**BINGO AND ENCODING-DECODING**

**(Functions and Arrays)**

**-Arpit Chandak (1941011)**

**-Rohith E P (1941037)**

**-Sourav Singha (1941047)**

**-Vedant Mehrotra (1941049)**

**Introduction to Functions and Arrays in C Programming-**

**Function-** A function is a set of statements that take inputs, do some specific computation and produces output.

* Functions help us in reducing code redundancy. If functionality is performed at multiple places in software, then rather than writing the same code, again and again, we create a function and call it everywhere. This also helps in maintenance as we have to change at one place if we make future changes to the functionality.
* Functions make code modular. Consider a big file having many lines of codes. It becomes really simple to read and use the code if the code is divided into functions.
* Functions provide abstraction. For example, we can use library functions without worrying about their internal working.

**Arrays-** An array is collection of items stored at continuous memory locations. Arrays are the derived data type in C programming language which can store the primitive type of data such as int, char, double, float, etc. It also has the capability to store the collection of derived data types, such as pointers, structure, etc. The array is the simplest data structure where each data element can be randomly accessed by using its index number.

For example, if we want to store the marks of a student in 6 subjects, then we don’t need to define different variables for the marks in the different subject. Instead of that, we can define an array which can store the marks in each subject at the contiguous memory locations.

ENCODING DECODING

USING THE CONCEPTS OF FUNCTIONS AND ARRAYS:

Encoding:

Here ,we encode a line of text which is taken as input from the user.

1. Each character , including blank spaces are converted into its ASCII equivalent.
2. A positive random integer is generated, which is added to the ASCII equivalent of each character.
3. The generated integer is the encoded value.

Decoding:

Here,we decode the encoded line of text which is generated in the encoding step.

1.Each encoded character, including the blank spaces are converted to their original decoded value.

2.Each encoded value is reduced by the generated random integer.

3.Hence ,we get the decoded value i.e the line of text given as input.

FUNCTION: ENCODING and DECODING is given in separate functions .

HEADER FILES USED-

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

**CODE-**

//convert the given line into ASCII

//generate a random no

//add this random no to ASCII of each character for encoding and the same no for decoding

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void enc(char \*,int,FILE \*);

void dec(char \*,int,FILE \*);

void ex();

char n[50];

void main()

{

system("COLOR B0");

int i,r,ch;

char a[50];

FILE \*fp;

while(1)

{

fp=fopen("encdec.txt","a");

printf("\n<--------------------------------------------------------------------------------------------------------------------->");

printf("\n\t\t\t\t\t\t1.Input a line\n\t\t\t\t\t\t2.Encode the line\n\t\t\t\t\t\t3.Decode the line\n\t\t\t\t\t\t4.Exit");

printf("\n\t\t\t\t\t\tEnter your choice: ");

scanf("%d",&ch);

printf("\n");

switch(ch)

{

case 1:

srand(time(NULL));

r = rand();

printf("\n\t\t\tEnter a sentence: ");

fflush(stdin);

gets(a);

break;

case 2:

enc(a,r,fp);

break;

case 3:

dec(a,r,fp);

break;

case 4:

ex();

default:

printf("\n\t\t\t\t\t\tEnter a valid choice!");

break;

}

}

}

void enc(char \*a,int r,FILE \*fp)

{

int i;

printf("\n\n\t\t\t\t\t\tGenerated random no: %d\n",r); //generated random number

for(i=0;i<strlen(a);i++)

{

n[i]=r+a[i]; //Encoding

}

printf("\n\t\t\t\t\tEncoded value::");

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

{

printf("%c",n[i]); //printing ASCII + random in char

}

fprintf(fp,"\n%s\n",n);

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

{

a[i]=n[i];

}

}

void dec(char \*a,int r,FILE \*fp)

{

int i;

printf("\n\t\t\t\t\t\tDecoded value::\n\t\t\t\t\t");

for(i=0;i<strlen(a);i++) //decoding

{

a[i]=a[i]-r;

printf("%c",a[i]); //printing the original sentence

}

fprintf(fp,"%s\n",a);

fclose(fp);

}

void ex()

{

printf("\n\t\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\t\t\t\t\t\t\* \*");

printf("\n\t\t\t\t\t\t\*SEE YOU SOON!\*");

printf("\n\t\t\t\t\t\t\* \*");

printf("\n\t\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n<--------------------------------------------------------------------------------------------------------------------->");

exit(1);

}

ALGORITHM-

Step 1: Print the options for the menu.

Step 2: Use switch case for the input.

Step 3: In the first case:

Read the input from the user.

Step 4: Generate a random number.

Step 5: Convert the input line into its ASCII equivalent to it and add the generated number to it to get the encoded value.

Step 6: Print this encoded value.

Step 7:Deduct the random integer from the encoded value, to get the decoded value.

Step 8: Print this decoded value in string.

Step 9: Allow the user to exit using exit function.

Bingo game

Bingo game using the concept of functions and arrays

Bingo game

In this game, the aim is to achieve bingo by satisfying the criteria of making certain number of rows and columns of the specified bingo table zero.

For example:

If the bingo table is of 3 rows and 3 columns then the combination of any 3 rows and any 3 columns must be 0 in order to achieve bingo.

If the table is of 4\*4 the number of combinations of rows and columns should be 4 for hitting a bingo

About the project

The program of this project uses the concept of arrays and functions

The program has two major parts

* Creating a bingo table
* Generating random numbers

Program logic

For the creation of bingo table and generating random numbers the program uses the concept of arrays and loops. The algorithm for this process is

* Input the number of rows and columns from the user
* Create a function to generate random numbers from 1 to 25
* Using these random numbers create a table with the specified number of rows columns
* Create another function to generate a single random number at every step of the game
* Print “row is filled” or “column is filled” whenever all the elements of the corresponding rows or columns becomes 0

* Print bingo when the number of rows or columns having 0 as all the elements in that particular row or column is equal to total number of rows and columns
* Allow the user to exit using the EXIT function

**CONCLUSION**

Hereby, with the completion of this project we learnt the concepts of arrays of functions and solving a real time based question. We learnt to encode and decode via their ASCII equivalent.

**Acknowledgement**

We would like to thank our lecturer Prof. Saleema J S for her support and guidance and we would also like to thank Christ University for providing us with the golden opportunity of completing a project in the 1st semester itself