PES 2019 Project 1 Readme

Utkarsh Dviwedi

The repository has been organized into project folders and program subfolders. There is a test folder for testing the git commit instructions. The Repository is public access. The description of the codes is given in the comments

Notes for running the code

Please enter a number on prompt for problem 2

**Program 2**

/\*

Utkarsh Dviwedi 2019

Principles of Embedded Software Fall 2019

Project 1 Program 2

Question

Write a program that uses a logical expression that tests whether a given character code is a

lower case [97-122]

upper case [65-90]

digit [48-57]

white space (like null, backspace, space, tabs, etc.) [0-15,20]

or a special character (like ! or >) in ASCII. [others less than 127]

References

https://stackoverflow.com/questions/6660145/convert-ascii-number-to-ascii-character-in-c

https://en.wikipedia.org/wiki/ASCII

\*/

#include <stdio.h>

#include <stdlib.h>

//Function Prototype

void TypeDetector (int Ascii,char c\_Ascii);

int main()

{

int Ascii,i;

char test;

printf("Enter the Ascii Value : ");

scanf("%d", &Ascii);

char c\_Ascii = Ascii;

TypeDetector(Ascii,c\_Ascii);

int Ascii\_Vals[] = {66,114,117,99,101,32,83,97,121,115,32,72,105,33,7,9,50,48,49,57,256};

int Vals\_count = ((sizeof(Ascii\_Vals))/(sizeof(Ascii\_Vals[0])));

printf("\n %d",Vals\_count);

for (i=0;i<=Vals\_count;i++)

{

c\_Ascii = Ascii\_Vals[i];

TypeDetector(Ascii\_Vals[i],c\_Ascii);

}

return 0;

}

//Function to Detect the Ascii symbol for an int

void TypeDetector (int Ascii,char c\_Ascii)

{

//For Lowercase Characters

if (Ascii >= 97 && Ascii <= 122 )

{

printf("\nCODE =%d Type = Lowercase Ascii Symbol =%c \n",Ascii,c\_Ascii);

}

//For digits

else if (Ascii >= 48 && Ascii <= 57 )

{

printf("\nCODE =%d Type = Number Ascii Symbol =%c \n",Ascii,c\_Ascii);

}

//For Uppercase Characters

else if (Ascii >= 65 && Ascii <= 90 )

{

printf("\nCODE =%d Type = Uppercase Ascii Symbol =%c \n",Ascii,c\_Ascii);

}

//For Invalid Characters

else if (Ascii > 127 )

{

printf("\nCODE =%d Type= Invalid character ASCII symbols have to between [0-126] \n",Ascii);

}

else if ( (Ascii== 20) || ((Ascii <= 15 && Ascii >=0 )) )

{

printf("\nCODE =%d Type= Space or Non Printable Character Ascii Symbol =%c \n",Ascii,c\_Ascii);

}

//For Special Characters

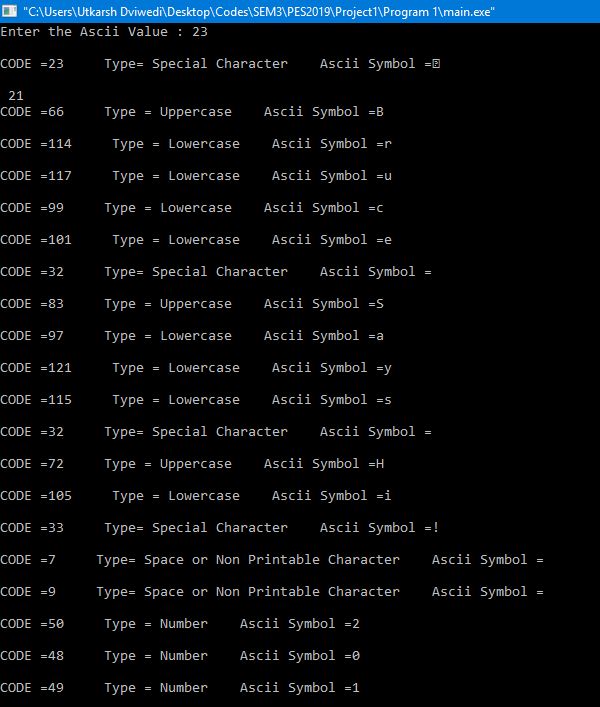
else

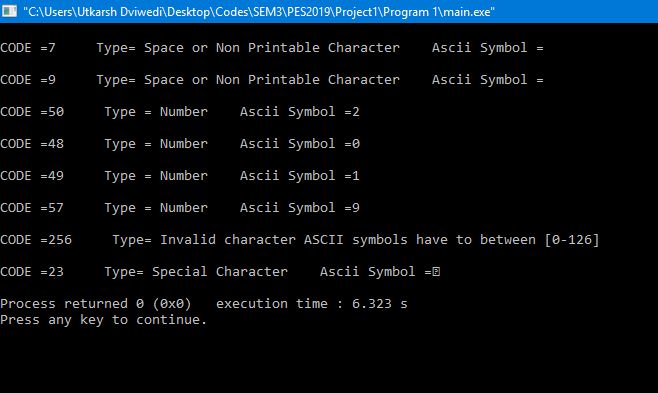
{

printf("\nCODE =%d Type= Special Character Ascii Symbol =%c \n",Ascii,c\_Ascii);

}

}





**Program 3**

/\*

Utkarsh Dviwedi 2019

Principles of Embedded Software Fall 2019

Project 1 Program 3

Question

Given the starting integer value 0xCAFE, perform each of these operations in series,

that is, each operation should be performed on the result of the previous function.

1 Print the results of each function to the command line (to capture as ProgramThree.out).

2 Print the original input in hexadecimal

3 Test if 3 of last 4 bits are on, and print the value in binary (along with the result of the test – true/false)

4 Reverse the byte order, print the value in hexadecimal

5 Test if 3 of last 4 bits are on, and print the value in binary (along with the result of the test – true/false)

6 Rotate the value by four bits to the left, print the value in hexadecimal

7 Test if 3 of last 4 bits are on, and print the value in binary (along with the result of the test – true/false)

8 Rotate the value by eight bits to the right, print the value in hexadecimal

9 Test if 3 of last 4 bits are on, and print the value in binary (along with the result of the test – true/false)

References

https://codeforwin.org/2015/08/c-program-to-convert-hexadecimal-to-bin\_tempary-number-system.html

https://www.tutorialspoint.com/cprogramming/c\_return\_arrays\_from\_function.htm

\*/

#include<stdlib.h>

#include <stdio.h>

#include <string.h> //For Strcat Functin

int bitsize = 15;

/////////////////////////////////

//Function Prototypes///////////

////////////////////////////////

//Function to Rotate an array(bintemp) left(Dir 0) or right(Dir 1) by index rotate value

char \*rotate(char bin\_temp[],int rotate\_val,int bitsize,int direction);

//Function to Reverse an Array.

char \*reversal(char bin\_temp[],int bitsize );

//Function to Convert 4 char bits to a Hex Number

char bit4\_binary\_to\_hex(char i,char j,char k,char l);

//Function to Print the

void PrintHex(char bin\_temp[]);

// Function to Check if three of the last four Bits are ON

void test3on(char bin[]);

// Function to Check if three of the last four Bits are ON

void test3on(char bin[]);

//Function to Convert a Hex Value to Binary

void HexToBin(char hex[],char \*bin);

//////////////////////////////////

//////////Main Function//////////

////////////////////////////////

int main()

{

char bin[17];

char \* binptr;

binptr =&bin;

char hex[] = "CAFE";

char \*Rev;

char \*Rotated1;

char \*Rotated2;

HexToBin(hex,binptr);

printf("\n Hexademial number = %s\n", hex);

printf("\n bin binary number = %s", bin);

test3on(binptr);

PrintHex(bin);

printf("\n Assuming all operations to be cascaded");

printf("\n\n ---------------- Reversal of the bits------------------------------------------------ ");

Rev = reversal(bin,bitsize);

printf("\n the Binary value now is %s",Rev);

test3on(Rev);

PrintHex(Rev);

printf("\n\n ---------------- Rotation 1 of the reversed bits by 4 bits to the left--------------- ");

Rotated1 = rotate(Rev,4,16,0);//Rotation by 4bits to the left

test3on(Rotated1);

PrintHex(Rotated1);

printf("\n\n -------------- Rotation 2 of previously rotated array by 8 bits to the right------ ");

Rotated2 = rotate(Rotated1,4,8,1);//Rotation by 8 bits to the Right

test3on(Rotated2);

PrintHex(Rotated2);

printf("\n\n-----------------Program Completed--------------------\n\n");

return 0;

}

///////////////////////////////////////

///////////Function Definitions////////

///////////////////////////////////////

//Function to Rotate an array(bintemp) left(Dir 0) or right(Dir 1) by index rotate value

char \*rotate(char bin\_temp[],int rotate\_val,int bitsize,int direction)

{

static char rotated[17];

int i;

rotated[16]='\0';

//Left Rotation

if (direction==0)

{

for (i=0;i<=bitsize-rotate\_val;i++)

{

rotated[i]=bin\_temp[i+rotate\_val];

}

for (i=bitsize-rotate\_val;i<bitsize;i++)

{

rotated[i]=bin\_temp[i-(bitsize-rotate\_val)];

}

printf("\n Original Number %s",bin\_temp);

printf("\n Number after Rotation %s",rotated);

}

//Right Rotation

if (direction==1)

{

rotate\_val = bitsize - rotate\_val;

for (i=0;i<=bitsize-rotate\_val;i++)

{

rotated[i]=bin\_temp[i+rotate\_val];

}

for (i=bitsize-rotate\_val;i<bitsize;i++)

{

rotated[i]=bin\_temp[i-(bitsize-rotate\_val)];

}

printf("\n Original Number %s",bin\_temp);

printf("\n Number after Rotation %s",rotated);

}

return rotated;

}

//Function to Reverse an Array

char \*reversal(char bin\_temp[],int bitsize )

{

static char reversed[17];

int i;

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

{

reversed[i] = bin\_temp[bitsize-i];

}

reversed[16]='\0';

return reversed;

}

//Function to Convert 4 char bits to a Hex Number

char bit4\_binary\_to\_hex(char i,char j,char k,char l)

{

if ( (i==48) && (j==48) && (k==48) && (l==48) )

{return '0';}

else if ( (i==48) && (j==48) && (k==48) && (l==49) )

{return '1';}

else if ( (i==48) && (j==48) && (k==49) && (l==48) )

{return '2';}

else if ( (i==48) && (j==48) && (k==49) && (l==49) )

{return '3';}

else if ( (i==48) && (j==49) && (k==48) && (l==48) )

{return '4';}

else if ( (i==48) && (j==49) && (k==48) && (l==49) )

{return '5';}

else if ( (i==48) && (j==49) && (k==49) && (l==48) )

{return '6';}

else if ( (i==48) && (j==49) && (k==49) && (l==49) )

{return '7';}

else if ( (i==49) && (j==48) && (k==48) && (l==48) )

{return '8';}

else if ( (i==49) && (j==48) && (k==48) && (l==49) )

{return '9';}

else if ( (i==49) && (j==48) && (k==49) && (l==48) )

{return 'A';}

else if ( (i==49) && (j==48) && (k==49) && (l==49) )

{return 'B';}

else if ( (i==49) && (j==49) && (k==48) && (l==48) )

{return 'C';}

else if ( (i==49) && (j==49) && (k==48) && (l==49) )

{return 'D';}

else if ( (i==49) && (j==49) && (k==49) && (l==48) )

{return 'E';}

else if ( (i==49) && (j==49) && (k==49) && (l==49) )

{return 'F';}

else

{return 'Z';}

}

//Function to Print the

void PrintHex(char bin\_temp[])

{

int i;

printf("\nThe Hex value after the operation is ");

for (i=0;i<=3;i++)

{

printf("%c",bit4\_binary\_to\_hex((int)bin\_temp[i\*4],(int)bin\_temp[(i\*4)+1],(int)bin\_temp[(i\*4)+2],(int)bin\_temp[(i\*4)+3]));

}

}

// Function to Check if three of the last four Bits are ON

void test3on(char bin[])

{

int count = 0;

int i = 0;

for (i = 12; i < 16; i++)

{

if (bin[i]=='1')

{

count++;

}

}

if (count == 3 )

{

printf("\n 3 of the last 4 bits are on");

}

else

{

printf("\n Statement that 3 of the last 4 bits are on is False ");

}

}

//Function to Convert a Hex Value to Binary

void HexToBin(char hex[],char \*bin)

{

char bin\_temp[65] = "";

//char bin[17];

int i = 0;

/\* Extract first digit and find bin\_temporary of each hex digit \*/

for(i=0; hex[i]!='\0'; i++)

{

switch(hex[i])

{

case '0':

strcat(bin\_temp, "0000");

break;

case '1':

strcat(bin\_temp, "0001");

break;

case '2':

strcat(bin\_temp, "0010");

break;

case '3':

strcat(bin\_temp, "0011");

break;

case '4':

strcat(bin\_temp, "0100");

break;

case '5':

strcat(bin\_temp, "0101");

break;

case '6':

strcat(bin\_temp, "0110");

break;

case '7':

strcat(bin\_temp, "0111");

break;

case '8':

strcat(bin\_temp, "1000");

break;

case '9':

strcat(bin\_temp, "1001");

break;

case 'a':

case 'A':

strcat(bin\_temp, "1010");

break;

case 'b':

case 'B':

strcat(bin\_temp, "1011");

break;

case 'c':

case 'C':

strcat(bin\_temp, "1100");

break;

case 'd':

case 'D':

strcat(bin\_temp, "1101");

break;

case 'e':

case 'E':

strcat(bin\_temp, "1110");

break;

case 'f':

case 'F':

strcat(bin\_temp, "1111");

break;

}

}

printf("\n binary number = %s", bin\_temp);

//Assign the binary value to

for (i=0;i<=15;i++)

{

bin[i]=bin\_temp[i];

}

bin[16]='\0';

printf("\n bin binary number = %s", bin);

}

