NIRAJ KUMAR

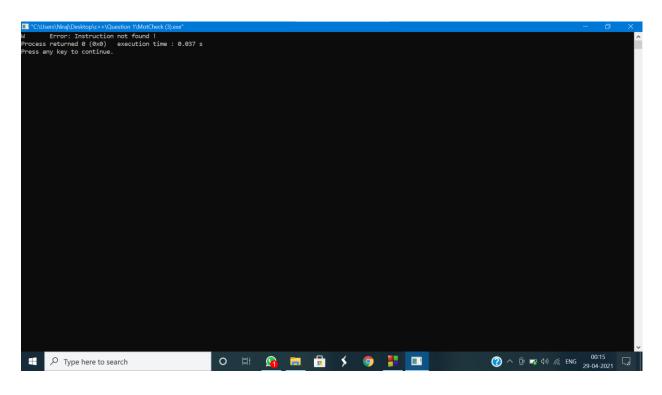
BT19CS031

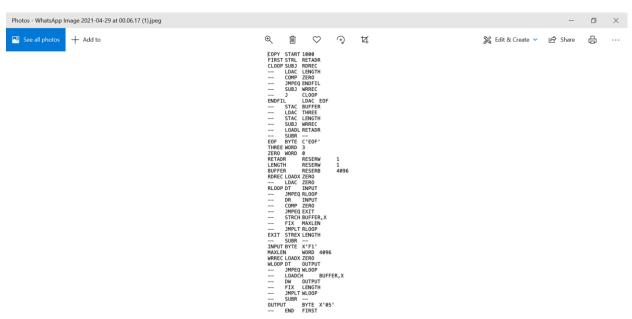
QUESTION 2

Extend the program implemented above to check whether the instructions present in the program is valid or not using a MOT table. If not present, then print an error message specifying that the particular instruction is not a valid instruction.

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
FILE *fpr=NULL, *fpw=NULL;
char addr(char ch)
    int n;
{
       n=ch-'0';
n=0;
        if(n==8)
n+=8;
return '8';
else if(n==9)
return '9';
else if(n==10)
return 'A';
else if(n==11)
return 'B';
else if(n==12)
return 'C';
else if(n==13)
return 'D';
else if(n==14)
return 'E';
else if(n==15)
return 'F';
```

```
} int
main() {
    char label[10],mnemonic[10]
,operand[10],program_name[10],sym_label[30][10],sym_address[30][10],
opcodes[1000][3],opcode[10],str[80],st[10],ch,c,chr[10],text[80][80];
int
address=0, line=0, length, x=0, first_address=0, last_address=0, pa=0, obj_code=0, i=
0, j=0, k=0, pos1=0,
    pos2=0, flaq1=0, flaq2=0, p1=0, p2=0, l=0, z=0, n=0, count=0, l1=0, l2=0;
char
op_mnemonic[26][10]={"LDAC", "STAC", "SUBJ", "MULT", "STRL", "DIVD", "ADDA", "STRCH"
,"DT",
"JMPEQ", "LOADCH", "DW", "SUBR", "FIXR", "FIX", "JMPLT", "COMP", "CLR", "LOADB", "J", "D
R", "COMPR", "STREX", "LOADT", "LOADL", "LOADX"};
char
op_opcode[26][10]={"10","1C","58","30","24","34","28","44","F0","40","60","EC
      "4C", "C8", "3C", "39", "38", "C4", "78", "3C", "E8", "D0", "20", "84", "08", "04"};
    fpr=fopen("instab.txt","r");
    fscanf(fpr, "%s %X\n", mnemonic, address);
line++;
            fclose(fpr);
    while(strcmp(mnemonic, "END")!=0)
    {
        for(i=0;i<26;i++){</pre>
             if(strcmp(mnemonic,op_mnemonic[i])==0){
break;
if(i==25){
                 printf("Error: Instruction not found !");
}
        printf("%s\n", mnemonic);
    return 0; }
```





.7