Name : Rapariya Dhruv Dineshbhai

Div : B

Roll No : 3162

Subject : SS (Practical)

Q-1

WAP to generate Symbol table for following:

$ START 101

$ MOVEM AREG A

LOOP MOVER AREG A

$ MOVER CREG B

$ BC ANY NEXT

NEXT SUB AREG A

LAST STOP

$ BC LT LOOP

A DS 1

B DS 1

BACK EQU LOOP

$ END

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include<ctype.h>
char mnemonic[3][3][10]=
{
    {"1","START","AD"},
    {"2","EQU","AD"}
};
char symbol_table[10][2][10]={""};
int s1=0;
int main()
{
    int i=0,j=0;
    int loc=0;
    int start=0,equ=0;
    char *field,record[200],const1[10];
    char symb_loc[25];
    int n;
    char op[20];
    FILE *fr;
    clrscr();
    printf("\n3162 Rapariya Dhruv D.\n");
    fr=fopen("C:\\TURBOC3\\SS\\ass_2.txt","r");
    while(!feof(fr))
    {
        int fcnt=0;
        loc++;
        fgets(record,200,fr);
    field=strtok(record," ");

        while(field!=NULL)
        {
            fcnt++;
            printf("%s \t",field);

            if(fcnt==1)
            {
                if(strcmp(field,"$")!=0)
                {
                    strcpy(symbol_table[s1][0],field);
                    strcpy(op,field);
                    sprintf(symb_loc,"%d",loc);
                    strcpy(symbol_table[s1][1],symb_loc);
                    s1++;
                }
            }
            if(fcnt==2)
            {
```

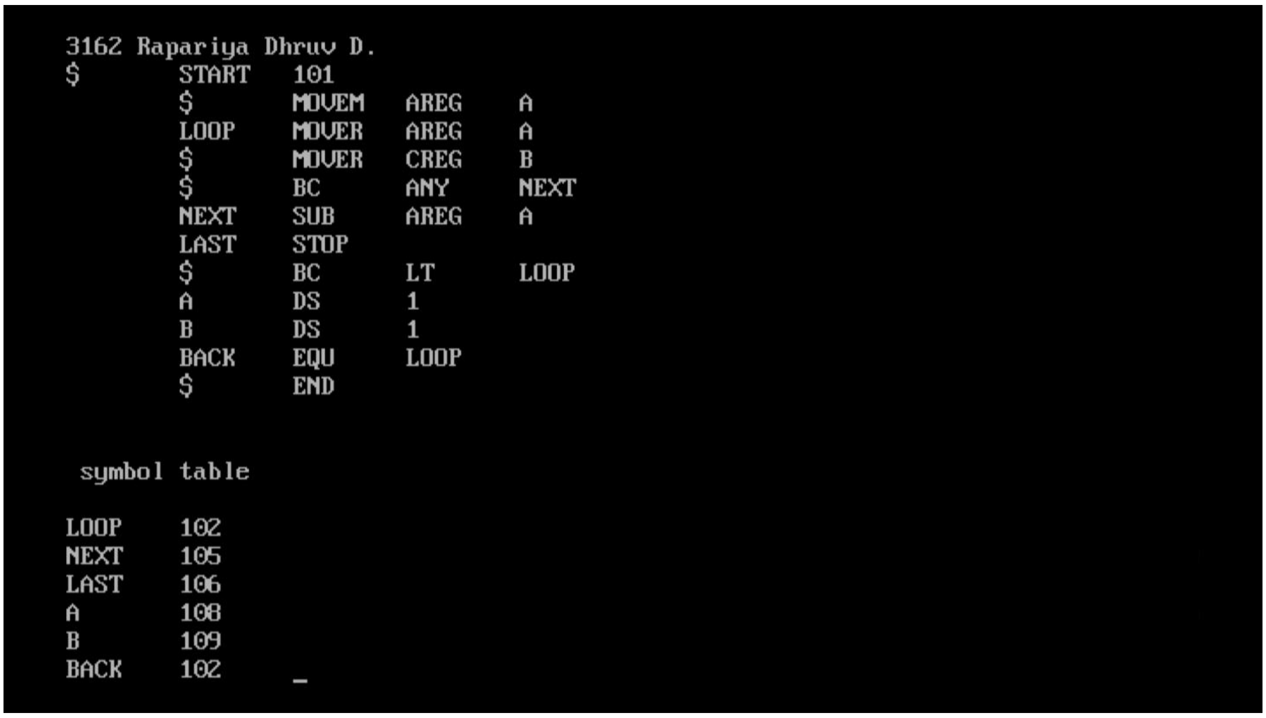
```
        int found=0;
        int index;
        for(i=0;i<3;i++)
        {
            if(strcmp(mnemonic[i][1],field)==0)
            {
                found=1;
                index=i;
                break;
            }
        }
        if(found==1)
        {
            char class1[10]="";
            char mnemonic1[10]="";
            strcpy(class1,mnemonic[index][2]);
            strcpy(mnemonic1,mnemonic[index][1]);
            if(strcmp(class1,"AD")==0)
            {
                if(strcmp(mnemonic1,"START")==0)
                {
                    start=1;
                }
                if(strcmp(mnemonic1,"EQU")==0)
                {
                    equ=1;
                    loc--;
                }
            }
        }
    }
    if(fcnt==3)
    {
        if(start==1)
        {
            strcpy(const1,field);
            loc=atoi(const1);
            loc=loc-1;
            start=0;
        }
        if(equ==1)
        {
            char index_of_symbol[20];
```

```
                int find_index=0;
                for(i=0;i<s1;i++)
                {
                    if(strcmp(symbol_table[i][0],field)==0)
                    {
                        if(strcmp(symbol_table[i][1]," ")!=0)
                        {
                            find_index=1;
                            strcpy(index_of_symbol,symbol_table[i][1]);
                            break;
                        }
                    }
                }
                if(find_index==1)
                {
                    for(i=0;i<s1;i++)
                    {
                        if(strcmp(symbol_table[i][0],op)==0)
                        {
                            strcpy(symbol_table[i][1],index_of_symbol);
                            break;
                        }
                    }
                    find_index=0;
                }
                equ=0;
            }
        }
        field=strtok(NULL," ");
    }
}
fclose(fr);

printf("\n \n \n symbol table\n");

for(i=0;i<s1;i++)
{
    printf("\n");
    for(j=0;j<2;j++)
    {
        printf("%s \t",symbol_table[i][j]);
    }
}
```

```
    getch();
    return 0;
}
```



```
3162 Rapariya Dhruv D.
$       START   101
        $       MOVEM   AREG    A
        LOOP    MOVER   AREG    A
        $       MOVER   CREG    B
        $       BC      ANY     NEXT
        NEXT    SUB     AREG    A
        LAST    STOP
        $       BC      LT      LOOP
        A       DS      1
        B       DS      1
        BACK    EQU     LOOP
        $       END


 symbol table

LOOP    102
NEXT    105
LAST    106
A       108
B       109
BACK    102     _
```

Q-2

WAP to generate Literal table and Pool table for following:

$ START 101

$ MOVER AREG =5

3162 Rapariya Dhruv Dineshbhai

```
$ MOVEM AREG A
LOOP MOVER AREG A
$ MOVER CREG B
$ ADD CREG =1
$ BC ANY NEXT
$ ORIGIN LOOP+1
NEXT SUB AREG A
$ LTORG
$ $ =5
$ $ =1
$ MOVER AREG =1
$ LTORG
$ $ =1
$ MOVER AREG =2
LAST STOP
$ BC LT BACK
A DS 1
B DS 1
BACK EQU LOOP
$ END
$ $ =2
```

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

char mnemonic[5][3][10]=
{
   {"1","START","AD"},
   {"2","EQU","AD"},
   {"3","ORIGIN","AD"},
   {"4","LTORG","AD"},
   {"5","END","AD"}
};

char symbol_table[10][2][10]={""};
char lit_table[10][2][10]={""};
int pool_table[10][2]={0};
int s1=0,l1=0,p1=0,l_cnt=0;

int main()
{
   int i=0,j;
   int loc=0;
   int start=0,equ=0,origin=0,ltorg=0,end=0;
   char *field,record[200],const1[10];
   char symb_loc[25];
   int n;
```

```
char op[20];
FILE *fr;
pool_table[0][0]=1;
pool_table[0][1]=0;
clrscr();
printf("\n3162 Rapariya Dhruv D.\n");
fr=fopen("C:\\TURBOC3\\SS\\ass_4.txt","r");

while(fgets(record,200,fr))
{
   int fcnt=0;   // field counter
   loc++;
//printf("\n");
   field=strtok(record," ");

   while(field!=NULL)
   {
      fcnt++;
      printf("%s \t",field);

      if(fcnt==1)
      {
   if(strcmp(field,"$")!=0)  // if field is not $ then label exist
         {
            strcpy(symbol_table[s1][0],field);
            strcpy(op,field);
            sprintf(symb_loc,"%d",loc);
            strcpy(symbol_table[s1][1],symb_loc);
            s1++;
        }//if not '$'
    }//if fcnt=1

    if(fcnt==2)
    {
        int found=0;
        int index;
        for(i=0;i<5;i++)
        {
            if(strcmp(mnemonic[i][1],field)==0)
            {
                found=1;
                index=i;
                break;
            }
        }
        if(found==1)
        {
            char class1[10]="";
            char mnemonic1[10]="";
            strcpy(class1,mnemonic[index][2]);
            strcpy(mnemonic1,mnemonic[index][1]);
            if(strcmp(class1,"AD")==0)
```

```
{
   if(strcmp(mnemonic1,"START")==0)
   {
      start=1;
   }
   if(strcmp(mnemonic1,"EQU")==0)
   {
      equ=1;
      loc--;
   }
   if(strcmp(mnemonic1,"ORIGIN")==0)
   {
      origin=1;
      loc--;
   }
   if(strcmp(mnemonic1,"LTORG")==0)
   {
      ltorg=1;
      loc--;
      break;
   }
   if(strcmp(mnemonic1,"END")==0)
   {
      end=1;
      loc--;
   }
```

```
        }
    }
}//if cnt=2
if(fcnt==3)
{
    if(start==1)
    {
        strcpy(const1,field);
        loc=atoi(const1);
        loc=loc-1;
        start=0;
    }
    if(equ==1)
    {
        char index_of_symbol[20];
        int find_index=0;
        for(i=0;i<s1;i++)
        {
            if(strcmp(symbol_table[i][0],field)==0)
            {
                if(strcmp(symbol_table[i][1]," ")!=0)
                {
                    find_index=1;
                    strcpy(index_of_symbol,symbol_table[i][1]);
                    break;
                }
        }
      }//for complete
      if(find_index==1)
      {
        for(i=0;i<s1;i++)
        {
          if(strcmp(symbol_table[i][0],op)==0)
          {
            strcpy(symbol_table[i][1],index_of_symbol);
            break;
          }
        }//for complete
        find_index=0;
      }//find_index =1 comlete
      equ=0;
    }   //if equ=1 complete
    if(origin==1)
    {
      char origin_str[20];
      char *p;
      char index_of_symbol[20];
      int find_index=0;
      strcpy(origin_str,field);
      p = strtok(origin_str, "+-");
      for(i=0;i<s1;i++)
      {
        if(strcmp(symbol_table[i][0],p)==0)
        {
            if(strcmp(symbol_table[i][1]," ")!=0)
            {
                find_index=1;
                strcpy(index_of_symbol,symbol_table[i][1]);
                break;
            }
        }
    }   //for complete
    if(find_index==1)
    {
        for(i=0;i<s1;i++)
        {
            if(strcmp(symbol_table[i][0],op)==0)
            {
                char *ptr = strchr(field, '+');
                p= (strtok(NULL, "+ -")) ;
                if(ptr)
                    loc= atoi(index_of_symbol)+atoi(p);
                else
                    loc=atoi(index_of_symbol)-atoi(p);
                sprintf(symb_loc,"%d",loc);
                break;
            }
        } // for complete
```

```
            find_index=0;
        }//find_index =1 comlete
        origin=0;
        loc--;
    }
    if(ltorg==1)
    {
        l_cnt++;
        if(l_cnt>l1)
        {
            ltorg=0;
            p1++;
            pool_table[p1][0]=l_cnt;
            pool_table[p1][1]=0;
            l_cnt--;
        }
        else
        {
            char *ptr;
            ptr=strchr(field,'=');
            if(ptr)
            {
                for(i=0;i<l1;i++)
                {
                    if(strcmp(lit_table[i][0],field)==0)
                    {
```

```
                    if(strcmp(lit_table[i][1]," ")==0)
                    {
                        sprintf(symb_loc,"%d",loc);
                        strcpy(lit_table[i][1],symb_loc);
                        pool_table[p1][1]=pool_table[p1][1]+1;
                    }
                }
            }
        }
    }
}
if(end==1)
{
    char *ptr;
    ptr=strchr(field,'=');
    if(ptr)
    {
        for(i=0;i<l1;i++)
        {
            if(strcmp(lit_table[i][0],field)==0)
            {
                if(strcmp(lit_table[i][1]," ")==0)
                {
                    sprintf(symb_loc,"%d",loc);
                    strcpy(lit_table[i][1],symb_loc);
                    pool_table[p1][1]=pool_table[p1][1]+1;
                    }
                  }
                }
              }
            }
          }//if fcnt=3

          if(fcnt==4)    // will write literals to littable /////
          {
            // int complete=0;
            // int lable_exist=0;
            char *ptr;
            ptr=strchr(field,'=');
            if(ptr)
            {
              strcpy(lit_table[l1][0],field);
              strcpy(lit_table[l1][1]," ");
              l1++;
            //   complete=1;
            }
          }
          field=strtok(NULL," ");
        }//while for all fields(tokens)
      }//eof while
      fclose(fr);
```

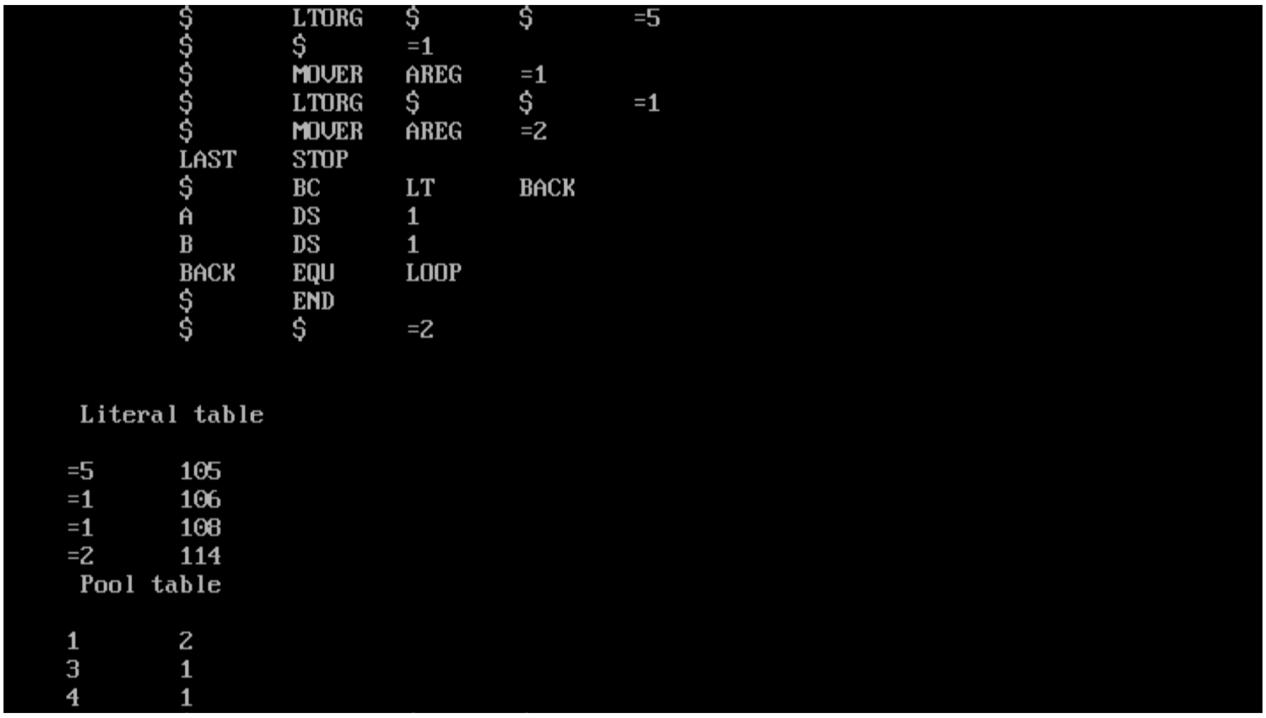
```
/*printf("\n \n \n Symbol table\n");

for(i=0;i<s1;i++)
{
printf("\n");
for(j=0;j<2;j++)
{
   printf("%s \t",symbol_table[i][j]);
}
} */

printf("\n \n \n Literal table\n");
for(i=0;i<l1;i++)
{
   printf("\n");
   for(j=0;j<2;j++)
   {
      printf("%s \t",lit_table[i][j]);
   }
}
printf("\n Pool table\n");

for(i=0;i<=p1;i++)
{
printf("\n");
for(j=0;j<2;j++)
```

```
   {
      printf("%d \t",pool_table[i][j]);
   }
   }
   getch();
   return 0;
}
```



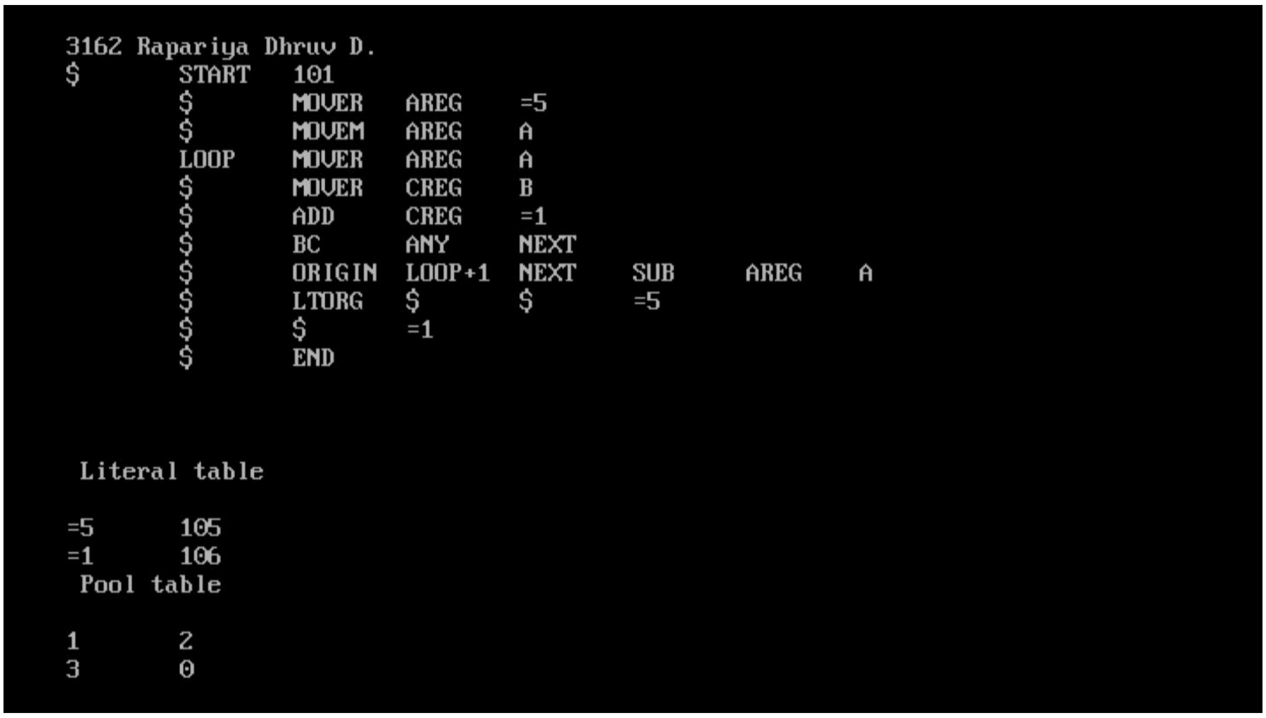
```
             $        LTORG    $        $        =5
             $        $        =1
             $        MOVER    AREG     =1
             $        LTORG    $        $        =1
             $        MOVER    AREG     =2
             LAST     STOP
             $        BC       LT       BACK
             A        DS       1
             B        DS       1
             BACK     EQU      LOOP
             $        END
             $        $        =2


      Literal table

     =5       105
     =1       106
     =1       108
     =2       114
      Pool table

     1        2
     3        1
     4        1
```

3162 **Rapariya Dhruv Dineshbhai**



```
3162 Rapariya Dhruv D.
$       START   101
        $       MOVER   AREG    =5
        $       MOVEM   AREG    A
        LOOP    MOVER   AREG    A
        $       MOVER   CREG    B
        $       ADD     CREG    =1
        $       BC      ANY     NEXT
        $       ORIGIN  LOOP+1  NEXT    SUB     AREG    A
        $       LTORG   $       $       =5
        $       $       =1
        $       END



 Literal table

=5      105
=1      106
 Pool table

1       2
3       0
```