PASS 1

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

void passOne(char label[10], char opcode[10], char operand[10],
char code[10], char mnemonic[3]);
void display();

int main()
{
   char label[10], opcode[10], operand[10];
   char code[10], mnemonic[3];
   passOne(label, opcode, operand, code, mnemonic);

   return 0;
}

void passOne(char label[10], char opcode[10], char operand[10], char code[10],
char mnemonic[3])
{
   int locctr, start, length;

   FILE *fp1, *fp2, *fp3, *fp4, *fp5;
   fp1 = fopen("input.txt", "r");
   fp2 = fopen("optab.txt", "r");
   fp3 = fopen("symtab.txt", "w");
   fp4 = fopen("intermediate.txt", "w");
   fp5 = fopen("length.txt", "w");

   fscanf(fp1, "%s\t%s\t%s", label, opcode, operand);

   if (strcmp(opcode, "START") == 0) {
      start = atoi(operand);
      locctr = start;
      fprintf(fp4, "\t%s\t%s\t%s\n", label, opcode, operand);
      fscanf(fp1, "%s\t%s\t%s", label, opcode, operand);
   }
   else {
      locctr = 0;
   }
   while (strcmp(opcode, "END") != 0) {
      fprintf(fp4, "%d\t%s\t%s\t%s\n", locctr, label, opcode, operand);
      if (strcmp(label, "**") != 0) {
         fprintf(fp3, "%s\t%d\n", label, locctr);
      }

      fscanf(fp2, "%s\t%s", code, mnemonic);
```

```
        while (strcmp(code, "END") != 0) {
            if (strcmp(opcode, code) == 0) {
                locctr += 3;
                break;
            }
            fscanf(fp2, "%s\t%s", code, mnemonic);
        }

        if (strcmp(opcode, "WORD") == 0) {
            locctr += 3;
        }
        else if (strcmp(opcode, "RESW") == 0) {
            locctr += (3 * (atoi(operand)));
        }
        else if (strcmp(opcode, "BYTE") == 0) {
            ++locctr;
        }
        else if (strcmp(opcode, "RESB") == 0) {
            locctr += atoi(operand);
        }
        fscanf(fp1, "%s\t%s\t%s", label, opcode, operand);
    }
    fprintf(fp4, "%d\t%s\t%s\t%s\n", locctr, label, opcode, operand);

    fclose(fp4);
    fclose(fp3);
    fclose(fp2);
    fclose(fp1);
    display();
    length = locctr - start;
    fprintf(fp5, "%d", length);
    fclose(fp5);
    printf("\nThe length of the code : %d\n", length);
}

void display() {

    char str;
    FILE *fp1, *fp2, *fp3;
    printf("\nThe contents of Input Table :\n\n");
    fp1 = fopen("input.txt", "r");
    str = fgetc(fp1);
    while (str != EOF) {
        printf("%c", str);
        str = fgetc(fp1);
    }
    fclose(fp1);
    printf("\n\nThe contents of Output Table :\n\n");
    fp2 = fopen("intermediate.txt", "r");
    str = fgetc(fp2);
    while (str != EOF) {
        printf("%c", str);
        str = fgetc(fp2);
    }
    fclose(fp2);
    printf("\n\nThe contents of Symbol Table :\n\n");
    fp3 = fopen("symtab.txt", "r");
    str = fgetc(fp3);
    while (str != EOF) {
        printf("%c", str);
        str = fgetc(fp3);
    }
    fclose(fp3);
}

/*

input.txt
---------

**      START   2000
**      LDA     FIVE
**      STA     ALPHA
**      LDCH    CHARZ
**      STCH    C1
ALPHA   RESW    2
FIVE    WORD    5
CHARZ           BYTE    C'Z'
C1      RESB    1
**      END     **


optab.txt
---------

LDA 03
STA     0f
LDCH    53
STCH    57
END     *

-----------------------------

symtab.txt
----------

ALPHA   2012
FIVE    2018
CHARZ   2021
C1      2022

intermediate.txt
----------------

      **    START   2000
2000  **    LDA     FIVE
2003  **    STA     ALPHA
2006  **    LDCH    CHARZ
2009  **    STCH    C1
2012  ALPHA RESW    2
2018  FIVE  WORD    5
2021  CHARZ BYTE    C'Z'
2022  C1    RESB    1
2023  **    END     **
```

```
//pass1 output

The contents of Input Table :

**      START   2000
**      LDA     FIVE
**      STA     ALPHA
**      LDCH    CHARZ
**      STCH    C1
ALPHA   RESW    2
FIVE    WORD    5
CHARZ   BYTE    C'Z'
C1      RESB    1
**      END     **


The contents of Output Table :

        **      START   2000
2000    **      LDA     FIVE
2003    **      STA     ALPHA
2006    **      LDCH    CHARZ
2009    **      STCH    C1
2012    ALPHA   RESW    2
2018    FIVE    WORD    5
2021    CHARZ   BYTE    C'Z'
2022    C1      RESB    1
2023    **      END     **


The contents of Symbol Table :

ALPHA   2012
FIVE    2018
CHARZ   2021
C1      2022

The length of the code : 23
```

PASS 2

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void display();
void swap(char *x, char *y) {
  char t = *x; *x = *y; *y = t;
}
char* reverse(char *buffer, int i, int j)
{
  while (i < j) {
    swap(&buffer[i++], &buffer[j--]);
  }
  return buffer;
}
char* itoa(int value, char* buffer, int base)
{
  if (base < 2 || base > 32) {
    return buffer;
  }
   int n = abs(value);
  int i = 0;
  while (n)
  {
    int r = n % base;
    if (r >= 10) {
      buffer[i++] = 65 + (r - 10);
    }
    else {
      buffer[i++] = 48 + r;
    }
    n = n / base;
  }
  if (i == 0) {
    buffer[i++] = '0';
  }
if (value < 0 && base == 10) {
    buffer[i++] = '-';
  }
  buffer[i] = '\0'; // null terminate string
  return reverse(buffer, 0, i - 1);
}
int main()
{
  char a[10], ad[10], label[10], opcode[10], operand[10], symbol[10];
  int start, diff, i, address, add, len, actual_len, finaddr, prevaddr, j = 0;
  char mnemonic[15][15] = {"LDA", "STA", "LDCH", "STCH"};
  char code[15][15] = {"33", "44", "53", "57"};
```

```
FILE *fp1, *fp2, *fp3, *fp4;
fp1 = fopen("output.txt", "w");
fp2 = fopen("symtab.txt", "r");
fp3 = fopen("intermediate.txt", "r");
fp4 = fopen("objcode.txt", "w");
fscanf(fp3, "%s\t%s\t%s", label, opcode, operand);
while (strcmp(opcode, "END") != 0)
{
  prevaddr = address;
  fscanf(fp3, "%d%s%s%s", &address, label, opcode, operand);
}
finaddr = address;
fclose(fp3);
fp3 = fopen("intermediate.txt", "r");
fscanf(fp3, "\t%s\t%s\t%s", label, opcode, operand);
if (strcmp(opcode, "START") == 0)
{
  fprintf(fp1, "\t%s\t%s\t%s\n", label, opcode, operand);
  fprintf(fp4, "H^%s^00%s^00%d\n", label, operand, finaddr);
  fscanf(fp3, "%d%s%s%s", &address, label, opcode, operand);
  start = address;
  diff = prevaddr - start;
  fprintf(fp4, "T^00%d^%d", address, diff);
}
while (strcmp(opcode, "END") != 0)
{
  if (strcmp(opcode, "BYTE") == 0)
  {
    fprintf(fp1, "%d\t%s\t%s\t%s\t", address, label, opcode, operand);
    len = strlen(operand);
    actual_len = len - 3;
    fprintf(fp4, "^");
    for (i = 2; i < (actual_len + 2); i++)
    {
      itoa(operand[i], ad, 16);
      fprintf(fp1, "%s", ad);
      fprintf(fp4, "%s", ad);
    }
    fprintf(fp1, "\n");
  }
  else if (strcmp(opcode, "WORD") == 0)
  {
    len = strlen(operand);
    itoa(atoi(operand), a, 10);
    fprintf(fp1, "%d\t%s\t%s\t%s\t00000%s\n", address, label, opcode, operand, a);
    fprintf(fp4, "^00000%s", a);
  }
  else if ((strcmp(opcode, "RESB") == 0) || (strcmp(opcode, "RESW") == 0)) {
```

```
            fprintf(fp1, "%d\t%s\t%s\t%s\n", address, label, opcode, operand);
        }
        else
        {
            while (strcmp(opcode, mnemonic[j]) != 0)
                j++;
            if (strcmp(operand, "COPY") == 0)
                fprintf(fp1, "%d\t%s\t%s\t%s\t%s0000\n", address, label, opcode, operand, code[j]);
            else
            {
                rewind(fp2);
                fscanf(fp2, "%s%d", symbol, &add);
                while (strcmp(operand, symbol) != 0)
                    fscanf(fp2, "%s%d", symbol, &add);
                fprintf(fp1, "%d\t%s\t%s\t%s\t%s%d\n", address, label, opcode, operand, code[j], add);
                fprintf(fp4, "^%s%d", code[j], add);
            }
        }
        fscanf(fp3, "%d%s%s%s", &address, label, opcode, operand);
    }
    fprintf(fp1, "%d\t%s\t%s\t%s\n", address, label, opcode, operand);
    fprintf(fp4, "\nE^00%d", start);
    fclose(fp4);
    fclose(fp3);
    fclose(fp2);
    fclose(fp1);
    display();
    return 0;
}
void display() {
    char ch;
    FILE *fp1, *fp2, *fp3, *fp4;
    printf("\nIntermediate file is converted into object code");
    printf("\n\nThe contents of Intermediate file:\n\n");
    fp3 = fopen("intermediate.txt", "r");
    ch = fgetc(fp3);
    while (ch != EOF)
    {
        printf("%c", ch);
        ch = fgetc(fp3);
    }
    fclose(fp3);
    printf("\n\nThe contents of Symbol Table :\n\n");
    fp2 = fopen("symtab.txt", "r");
    ch = fgetc(fp2);
    while (ch != EOF)
    {
        printf("%c", ch);
```

```
ch = fgetc(fp2);
  }
  fclose(fp2);
  printf("\n\nThe contents of Output file :\n\n");
  fp1 = fopen("output.txt", "r");
  ch = fgetc(fp1);
  while (ch != EOF)
  {
    printf("%c", ch);
    ch = fgetc(fp1);
  }
  fclose(fp1);
  printf("\n\nThe contents of Object code file :\n\n");
  fp4 = fopen("objcode.txt", "r");
  ch = fgetc(fp4);
  while (ch != EOF)
  {
    printf("%c", ch);
    ch = fgetc(fp4);
  }
  fclose(fp4);

}
```

intermediate.txt

```
----------------

      **    START   2000
2000   **    LDA    FIVE
2003   **    STA    ALPHA
2006   **    LDCH   CHARZ
2009   **    STCH   C1
2012   ALPHA  RESW   2
2018   FIVE   WORD   5
2021   CHARZ  BYTE   C'Z'
2022   C1     RESB   1
2023   **    END    **
```

output.txt

```
----------

      **    START   2000
2000   **    LDA    FIVE   332018
2003   **    STA    ALPHA  442012
2006   **    LDCH   CHARZ  532021
2009   **    STCH   C1     572022
2012   ALPHA  RESW   2
2018   FIVE   WORD   5     000005
2021   CHARZ  BYTE   C'Z'   5a
2022   C1     RESB   1
2023   **    END    **
```

symtab.txt

```
----------

ALPHA   2012
FIVE    2018
CHARZ   2021
C1      2022

-----------------------------
```

objcode.txt

```
-----------

H^***^002000^002023
T^002000^22^332018^442012^532021^572022^000005^5a
E^002000
```

```
//pass2 output

Intermediate file is converted into object code

The contents of Intermediate file:

        **      START   2000
2000    **      LDA     FIVE
2003    **      STA     ALPHA
2006    **      LDCH    CHARZ
2009    **      STCH    C1
2012    ALPHA   RESW    2
2018    FIVE    WORD    5
2021    CHARZ   BYTE    C'Z'
2022    C1      RESB    1
2023    **      END     **


The contents of Symbol Table :

ALPHA   2012
FIVE    2018
CHARZ   2021
C1      2022


The contents of Output file :

        **      START   2000
2000    **      LDA     FIVE    332018
2003    **      STA     ALPHA   442012
2006    **      LDCH    CHARZ   532021
2009    **      STCH    C1      572022
2012    ALPHA   RESW    2
2018    FIVE    WORD    5       000005
2021    CHARZ   BYTE    C'Z'    5A
2022    C1      RESB    1
2023    **      END     **


The contents of Object code file :

H^**^002000^002023
T^002000^22^332018^442012^532021^572022^000005^5A
E^002000
```

Absolute-Loader

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
void main() {
  FILE * fp;
  int i, addr1, l, j, staddr1;
  char name[10], line[50], name1[10], addr[10], rec[10], ch, staddr[10];
  printf("enter program name:");
  scanf("%s", name);
  fp = fopen("input.txt", "r");
  fscanf(fp, "%s", line);
  for (i = 2, j = 0; i < 8, j < 6; i++, j++)
    name1[j] = line[i];
  name1[j] = '\0';
  printf("name from obj.%s\n", name1);
  if (strcmp(name, name1) == 0) {
    do {
      fscanf(fp, "%s", line);
      if (line[0] == 'T') {
        for (i = 2, j = 0; i < 8, j < 6; i++, j++)
          staddr[j] = line[i];
        staddr[j] = '\0';
        staddr1 = atoi(staddr);
        i = 12;
        while (line[i] != '$') {
          if (line[i] != '^') {
            printf("00%d \t %c%c\n",
              staddr1, line[i], line[i + 1]);
            staddr1++;
            i = i + 2;
          } else i++;
        }
      } else if (line[0] = 'E') {
        fclose(fp);
        exit(0);
      }
    } while (!feof(fp));
  }
}
```

```
//absolute loader output

Enter the name of the program:FIRST
name from obj File. FIRST
002000   03
002001   20
002002   18
002003   0f
002004   20
002005   12
002006   53
002008   21
002009   57
002010   20
002011   22
002012   00
002013   00
002014   05
002015   5a

Execution address: 002000
```

```
//objcode.txt

H^FIRST^002000^002023
T^002000^22^032018^0f2012^532021^572022^000005^5a
E^002000
```

Relocating-Loader

```
#include<stdio.h>
#include<string.h>
#include <stdlib.h>
char bit[30];
char bitmask[20];
void bitmask_convert(char mask[])
{
    int len;
    len=strlen(mask);
    strcpy(bit,"");
    int i;
    for(i=0;i<len;++i)
    {
        switch(mask[i])
        {

            case '0': strcat(bit,"0000");
                      break;
            case '1': strcat(bit,"0001");
                      break;
            case '2': strcat(bit,"0010");
         break;
            case '3': strcat(bit,"0011");
         break;
            case '4': strcat(bit,"0100");
         break;
            case '5': strcat(bit,"0101");
         break;
            case '6': strcat(bit,"0110");
         break;
            case '7': strcat(bit,"0111");
         break;
            case '8': strcat(bit,"1000");
         break;
            case '9': strcat(bit,"1001");
         break;
            case 'A': strcat(bit,"1010");
         break;
            case 'B': strcat(bit,"1011");
         break;
            case 'C': strcat(bit,"1100");
         break;
            case 'D': strcat(bit,"1101");
         break;
            case 'E': strcat(bit,"1110");
         break;
            case 'F': strcat(bit,"1111");
```

```
                  break;
                        default : break;
                }
        }
}
void main()
{
        FILE *objptr;
        int start,addr;
        char rec[20];
        char name[20];
        int modif_obj_code;
        char first[3];
        char second[5];
        int bitmask_index=0;

        int i;
        int  add,len;
        printf("ENTER THE STARTING ADDRESS OF THE PROGRAM\n");
        scanf("%X",&start);
        addr=start;
        objptr=fopen("program.txt","r");
    fscanf(objptr,"%s",rec);
    if(strcmp(rec,"H")==0)
    {
            fscanf(objptr,"%s",name);
            fscanf(objptr,"%X",&add);
            fscanf(objptr,"%X",&len);
           printf("\nPROGRAM NAME=%s\n\n",name);
           printf(" ADDRESS   OBJECT CODE \n");
           printf("______________________________\n");
    }
    else
        {
                printf("INAVLID OBJECT CODE FORMAT\n");
                fclose(objptr);
                exit(1);
        }
    strcpy(rec,"");
    fscanf(objptr,"%s",rec);
    while(strcmp(rec,"E")!=0)
    {
    if(strcmp(rec,"T")==0)
            {
                    fscanf(objptr,"%X",&add);
                    fscanf(objptr,"%X",&len);
                    fscanf(objptr,"%s",bitmask);
                    add+=start;
                    bitmask_index=0;
            bitmask_convert(bitmask);
            fscanf(objptr,"%s",rec);
            }
                if(bit[bitmask_index]=='1')
                {
                        for(i=0;i<6;++i)
                        {
                                if(i<2)
                                {

                                first[i]=rec[i];

                                }
                                else
                                {
                                        second[i-2]=rec[i];
                                }
                        }
                        first[2]='\0';
                        second[4]='\0';
                        modif_obj_code=strtol(second,NULL,16);
                        modif_obj_code+=start;

                        printf("%X\t%s%X\n",add,first,modif_obj_code);
                }
                else
                {
                        printf("%X\t%s\n",add,rec);

                }
                add+=3;
                bitmask_index++;
                   fscanf(objptr,"%s",rec);
    }
   fclose(objptr);
}
```

```
//relocation loader output


Enter the actual starting address : 5000

 The contents of output file(ROutput.txt)
 ----------------------------
 ADDRESS         CONTENT
 ----------------------------

  5000            145033

  5003            486039

  5006            105036

  5009            285030

  500c            305015

  500f            486061

  5012            3c5003

  5015            20502a

  5018            1c5039

  501b            30502d

  7500            1d5036

  7503            486061

  7506            185033

  7509            4c1000

  750c            801000

  750f            601003
----------------------------
```

```
//RInput.txt

H COPY 000000 00107A
T 000000 1E FFC 14 0033 48 1039 10 0036 28 0030 30 0015 48 1061 3C 0003 20 002A 1C 0039 30 002D
T 002500 15 E00 1D 0036 48 1061 18 0033 4C 1000 80 1000 60 1003
E 000000
```

```
//ROutput.txt
------------------------------
ADDRESS      CONTENT
------------------------------

5000            145033

5003            486039

5006            105036

5009            285030

500c            305015

500f            486061

5012            3c5003

5015            20502a

5018            1c5039

501b            30502d

7500            1d5036

7503            486061

7506            185033

7509            4c1000

750c            801000

750f            601003
------------------------------
```