/*NAME : BHAGYA A JAI
ROLL NO : B21CSB18
PASS 1*/

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<stdbool.h>
FILE *F1, *F2, *F3, *F4, *F5 ;
typedef struct OPTAB{
char operand[10] ;
char opcode[10] ;
}OPTAB ;
OPTAB optable[10] ;
int count_optable = 0 ;
typedef struct SYMTAB{
char symbol[10] ;
int address ;
}SYMTAB ;
SYMTAB symtable[10] ;
int count_symtable = 0 ;
char LABEL[10], OPCODE[10], OPERAND[10] ;
int starting_address ;
int locctr ;
void read_optab(){
F2 = fopen("OPTAB.txt", "r") ;
while(true){
fscanf(F2, "%s\t%s", optable[count_optable].operand,
optable[count_optable].opcode) ;
if(fgetc(F2) == EOF){
break ;
}
count_optable++ ;
}
fclose(F2) ;
}
void read_line(){
char t1[10] ;
char t2[10] ;
char t3[10] ;
strcpy(t1, "") ;
strcpy(t2, "") ;
strcpy(t3, "") ;
fscanf(F1, "%s", t1) ;
if(getc(F1) != '\n'){
fscanf(F1, "%s", t2) ;
if(getc(F1) != '\n'){
fscanf(F1, "%s", t3) ;
}
}
if(strcmp(t1, "") != 0 && strcmp(t2, "") != 0 && strcmp(t3, "") != 0){
strcpy(LABEL, t1) ;
strcpy(OPCODE, t2) ;
strcpy(OPERAND, t3) ;
```

```
}
else if(strcmp(t1, "") != 0 && strcmp(t2, "") != 0 && strcmp(t3, "") == 0){
strcpy(LABEL, "") ;
strcpy(OPCODE, t1) ;
strcpy(OPERAND, t2) ;
}
else if(strcmp(t1, "") != 0 && strcmp(t2, "") == 0 && strcmp(t3, "") == 0){
strcpy(LABEL, "") ;
strcpy(OPCODE, t1) ;
strcpy(OPERAND, "") ;
}
}
void main(){
F1 = fopen("INPUT.txt", "r") ;
F3 = fopen("SYMTAB.txt", "w") ;
F4 = fopen("INTERMEDIATE.txt", "w") ;
F5 = fopen("LENGTH.txt", "w") ;
read_optab() ;
fscanf(F1, "%s\t%s\t%X", LABEL, OPCODE, &locctr) ;
if(strcmp(OPCODE, "START") == 0){
starting_address = locctr ;
printf("%X\n", starting_address) ;
locctr = starting_address ;
}
else{
locctr = 0 ;
starting_address = 0 ;
}
fprintf(F4, "\t%s\t%s\t%X\n", LABEL, OPCODE, locctr) ;
read_line() ;
while(strcmp(OPCODE, "END") != 0){
if(strcmp(LABEL, "") != 0){
for(int i = 0 ; i < count_symtable ; i++){
if(strcmp(symtable[i].symbol, LABEL) == 0){
printf("DUPLICATE ENTRY FOUND\n") ;
exit(0) ;
}
}
strcpy(symtable[count_symtable].symbol, LABEL) ;
symtable[count_symtable].address = locctr ;
count_symtable++ ;
fprintf(F3, "%s\t%X\n", LABEL, locctr) ;
}
fprintf(F4, "%X\t%s\t%s\t%s\n", locctr, LABEL, OPCODE, OPERAND) ;
int found = 0 ;
for(int i = 0 ; i < count_optable ; i++){
if(strcmp(optable[i].operand, OPCODE) == 0){
found = 1 ;
locctr += 0X3 ;
break ;
}
}
if(!found){
if(strcmp(OPCODE, "WORD") == 0){
locctr += 0X3 ;
```

```
}
else if(strcmp(OPCODE, "RESW") == 0){
locctr += 0X3 * atoi(OPERAND) ;
}
else if(strcmp(OPCODE, "RESB") == 0){
locctr += atoi(OPERAND) ;
}
else if(strcmp(OPCODE, "BYTE") == 0){
int len = strlen(OPERAND) ;
if(OPERAND[0] == 'C' || OPERAND[0] == 'c'){
len -= 3 ;
}
else{
len = (len - 3) / 2 ;
}
locctr += len ;
}
}
read_line() ;
}
fprintf(F4, "\t\t%s\t%X", OPCODE, starting_address) ;
fprintf(F5, "%X", locctr - starting_address) ;
fclose(F1) ;
fclose(F3) ;
fclose(F4) ;
fclose(F5) ;
}
```

INPUT

INPUT.txt

```
PGM1 START 1000
LDA ALPHA
MUL BETA
STA GAMMA
ALPHA WORD 2
BETA WORD 4
GAMMA RESW 1
 END 1000
```

OPTAB.txt

```
LDA 00
MUL 20
STA 0C
```

OUTPUT

SYMTAB.txt

```
ALPHA 1009
BETA 100C
GAMMA 100F
```

LENGTH.txt

```
12
```

INTERMEDIATE.txt

```
PGM1 START 1000
1000 LDA ALPHA
1003 MUL BETA
1006 STA GAMMA
1009 ALPHA WORD 2
100C BETA WORD 4
100F GAMMA RESW 1
END 1000
```

/*NAME : BHAGYA A JAI
ROLL NO : B21CSB18
PASS 2*/

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<stdbool.h>
typedef struct OPTAB{
char operation[10] ;
char opcode[10] ;
}OPTAB ;
OPTAB optable[10] ;
int count_optable = 0 ;
typedef struct SYMTAB{
char symbol[30] ;
char address[10] ;
}SYMTAB ;
SYMTAB symtable[10] ;
int count_symtable = 0 ;
typedef struct TEXT_RECORD{
char temp_entry[10] ;
}TEXT_RECORD ;
TEXT_RECORD temp_text_record[10] ;
int programlength ;
void read_length(){
FILE *F = fopen("LENGTH.txt", "r") ;
fscanf(F, "%X", &programlength) ;
fclose(F) ;
}
void read_opcode_table(){
FILE *F = fopen("OPTAB.txt", "r") ;
while(true){
fscanf(F, "%s\t%s", optable[count_optable].operation,
optable[count_optable].opcode) ;
if(fgetc(F) == EOF){
break ;
}
count_optable++ ;
}
fclose(F) ;
}
void read_symbol_table(){
FILE *F = fopen("SYMTAB.txt", "r") ;
while(true){
fscanf(F, "%s\t%s", symtable[count_symtable].symbol,
symtable[count_symtable].address) ;
if(fgetc(F) == EOF){
break ;
}
count_symtable++ ;
}
fclose(F) ;
}
```

```
FILE *F1, *F2 ;
char LABEL[10], OPCODE[10], OPERAND[10], LOCCTR[10] ;
int starting_address ;
void read_line(){
char t1[10], t2[10], t3[10], t4[10] ;
strcpy(t1, "") ;
strcpy(t2, "") ;
strcpy(t3, "") ;
strcpy(t4, "") ;
fscanf(F1, "%s", t1) ;
if(getc(F1) != '\n'){
fscanf(F1, "%s", t2) ;
if(getc(F1) != '\n'){
fscanf(F1, "%s", t3) ;
if(getc(F1) != '\n'){
fscanf(F1, "%s", t4) ;
}
}
}
if(strcmp(t1, "") != 0 && strcmp(t2, "") != 0 && strcmp(t3, "") != 0 &&
strcmp(t4, "") != 0){
strcpy(LOCCTR, t1) ;
strcpy(LABEL, t2) ;
strcpy(OPCODE, t3) ;
strcpy(OPERAND, t4) ;
}
else if(strcmp(t1, "") != 0 && strcmp(t2, "") != 0 && strcmp(t3, "") != 0 &&
strcmp(t4, "") == 0){
strcpy(LOCCTR, t1) ;
strcpy(LABEL, "") ;
strcpy(OPCODE, t2) ;
strcpy(OPERAND, t3) ;
}
else if(strcmp(t1, "") != 0 && strcmp(t2, "") != 0 && strcmp(t3, "") == 0 &&
strcmp(t4, "") == 0){
if(strcmp(t1, "END") == 0){
strcpy(LOCCTR, "") ;
strcpy(LABEL, "") ;
strcpy(OPCODE, t1) ;
strcpy(OPERAND, t2) ;
}
else{
strcpy(LOCCTR, t1) ;
strcpy(LABEL, "") ;
strcpy(OPCODE, t2) ;
strcpy(OPERAND, "") ;
}
}
}
void main(){
F1 = fopen("INTERMEDIATE.txt", "r") ;
F2 = fopen("OBJECT.txt", "w") ;
read_length() ;
read_opcode_table() ;
read_symbol_table() ;
```

```
fscanf(F1, "%s\t%s\t%s", LABEL, OPCODE, OPERAND) ;
fprintf(F2, "H^00%s^00%s^0000%X\n", LABEL, OPERAND, programlength) ;
read_line() ;
int curlen = 0X0 ;
char recordStartAddress[10] ;
printf("%X\n", programlength) ;
int iter = 0 ;
while(strcmp(OPCODE, "END") != 0){
if(curlen == 0){
strcpy(recordStartAddress, LOCCTR) ;
}
if(strcmp(OPCODE, "WORD") == 0){
int val = atoi(OPERAND) ;
if(val < 10){
strcpy(temp_text_record[iter].temp_entry,"00000") ;
strcat(temp_text_record[iter].temp_entry, OPERAND) ;
}
else if(val >= 10){
strcpy(temp_text_record[iter].temp_entry,"0000") ;
strcat(temp_text_record[iter].temp_entry, OPERAND) ;
}
curlen += 0X3 ;
iter++ ;
}
else if(strcmp(OPCODE, "BYTE") == 0){
int currlen = 0 ;
char tempArray[10] ;
FILE *F3 = fopen("ASSEMBLY_GENERATOR.txt", "w") ;
for(int j = 2 ; OPERAND[j + 1] != '\0' ; j++){
fprintf(F3, "%02X", OPERAND[j]) ;
curlen += 0X1 ;
}
fclose(F3) ;
F3 = fopen("ASSEMBLY_GENERATOR.txt", "r") ;
fscanf(F3, "%s", tempArray) ;
strcpy(temp_text_record[iter].temp_entry,tempArray) ;
fclose(F3) ;
iter++ ;
}
else if(strcmp(OPCODE, "RESB") == 0 || strcmp(OPCODE, "RESW") == 0){
//do nothing
}
else{
int k = 0 ;
int l = 0 ;
while(strcmp(optable[k].operation, OPCODE) != 0){
k++ ;
}
while(strcmp(symtable[l].symbol, OPERAND) != 0){
l++ ;
}
strcpy(temp_text_record[iter].temp_entry, optable[k].opcode) ;
strcat(temp_text_record[iter].temp_entry, symtable[l].address) ;
curlen += 0X3 ;
iter++ ;
```

```
}
printf("%d\n", curlen ) ;
if(curlen == 30 || curlen == 29 || curlen == 28){
fprintf(F2, "T^%X", curlen) ;
for(int i = 0 ; i < iter ; i++){
fprintf(F2, "^%s", temp_text_record[i].temp_entry) ;
}
fprintf(F2, "\n") ;
curlen = 0 ;
iter = 0 ;
}
read_line() ;
}
fprintf(F2, "T^00%s^%X", recordStartAddress, curlen) ;
for(int i = 0 ; i < iter ; i++){
fprintf(F2, "^%s", temp_text_record[i].temp_entry) ;
}
fprintf(F2, "\n") ;
fprintf(F2, "E^00%s\n", OPERAND) ;
fclose(F1) ;
fclose(F2) ;
}
```

INPUT

OPTAB.txt

```
LDA 00
MUL 20
STA 0C
```

LENGTH.txt

```
12
```

SYMTAB.txt

```
ALPHA 1009
BETA 100C
GAMMA 100F
```

INTERMEDIATE.txt

```
PGM1 START 1000
1000 LDA ALPHA
1003 MUL BETA
1006 STA GAMMA
1009 ALPHA WORD 2
100C BETA WORD 4
100F GAMMA RESW 1
END 1000
```

OUTPUT

OBJECT.txt

```
H^00PGM1^001000^000012
T^001000^F^001009^20100C^0C100F^000002^000004
E^001000
```

/*NAME : BHAGYA A JAI
ROLL NO : B21CSB18
ABSOLUTE LOADER*/

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
FILE *F1, *F2, *F3 ;
char line[100], addr[10] ;
int hexAddr ;
void main(){
int i, j ;
F1 = fopen("OBJECT.txt", "r+") ;
fscanf(F1, "%s", line) ;
printf("Program in memory :-\n") ;
j = 0 ;
for(int k = 9 ; k < 15 ; k++){
addr[j++] = line[k] ;
}
FILE *F2 = fopen("TEMP.txt", "r+") ;
fprintf(F2, "%s", addr) ;
rewind(F2) ;
fscanf(F2, "%X", &hexAddr) ;
fclose(F2) ;
do{
fscanf(F1, "%s", line) ;
if(line[0] == 'T'){
if(line[10] == '^'){
i = 11 ;
}
else{
i = 12 ;
}
while(line[i] != '\0'){
printf("%X:", hexAddr) ;
for(int k = 0 ; k < 6 ; k++){
printf("%c", line[i++]) ;
}
printf("\n") ;
if(line[i] == '\0'){
break ;
}
i++ ;
hexAddr += 0X3 ;
}
}
}while(line[0] != 'E') ;
fclose(F1) ;
}
```

INPUT

OBJECT.txt

```
H^00PGM1^001000^000012
T^001000^F^001009^20100C^0C100F^000002^000004
E^001000
```

OUTPUT

Program in memory
1000:001009
1003:20100C
1006:0C100F
1009:000002
100C:000004

/*NAME : BHAGYA A JAI
ROLL NO : B21CSB18
RELOCATING LOADER*/

PROGRAM

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
FILE *obj;
struct mod{

char op[10];
 int reloc[10];
 int flag[10];
}r;
char prog_name[10], record[100], locn[6], instr[2], objcode[9], cnt[2], relocn[6];
int i, j, k = 0, flag = 0, rec_len, start, ind, new_loc, load_addr, count = 0,
mod_count = 0, minstr;
int rel_addr;
void main(){
obj = fopen("OBJ.txt", "r");
 printf("Enter load address:- ");
 scanf("%X", &load_addr);
 for(i = 5, k = 0; k < 4, i <= 8; k++, i++){

locn[k] = record[i];

}
while(record[0] != 'E'){

if(record[0] == 'M'){

for(i = 2, j = 0; i < 8, j < 6; i++, j++){

relocn[j] = record[i];

}
sscanf(relocn, "%X", &rel_addr);
rel_addr += load_addr;
r.reloc[mod_count] = rel_addr;
r.op[mod_count] = record[11];
 r.flag[mod_count] = 0;
 mod_count ++;

}
fscanf(obj, "%s", record);
 }
rewind(obj);
 fscanf(obj, "%s", record);
 printf("Location Object Code\n");
 while(record[0] != 'E'){

if(record[0] == 'T'){

for (j = 4, k = 0; j < 8, k < 4; k++, j++){
```

```
 locn[k] = record[j];

}
sscanf(locn, "%X", &start);
 new_loc = start ;
 new_loc += load_addr ;
for(i=9, k=0; i<11, k<2; i++, k++){

cnt[k] = record[i];

}
sscanf(cnt, "%x", &count);
 count = count / 3;
 ind = 12;
 for(i = 0; i < 10; i++){

while(count > 0){

objcode[0] = '\0';
 for(j = 0, k = ind; j < 6, k < ind + 6; j++, k+
+){

objcode[j] = record[k];

}
 ind += 6 ;
if(record[ind] == '^' || record[ind] == '\0'){

ind++ ;
 objcode[j] = '\0';

}
 else{

while(record[ind] != '^'){

if(record[ind] == '\0'){

break;

}
 objcode[j] = record[ind] ;
 ind++;
 j++;

}
ind++;
 }
objcode[j] == '\0';
 sscanf(objcode, "%x", &minstr) ;
 for(i = 0; i < mod_count; i++){

if(r.reloc[i] > new_loc && r.reloc[i] <
new_loc+4 && r.flag[i] != 1){
```

```
if(r.op[i] == '+'){

minstr += load_addr;
 }
 else{

minstr -= load_addr;

}
 r.flag[i] = 1 ;
 break;

}

}
printf("%X\t%X\n",new_loc,minstr) ;
 new_loc += strlen(objcode) / 2;
 count--;
}
}
}
 fscanf(obj, "%s", record);

}
}
```

INPUT

OBJECT.txt

```
H^COPY^000000^000030
T^000000^1C^17202D^69202D^4B101036^032026^290000^332007^4B10105D^3F3FEC^032010
T^00001C^13^0F2016^010003^0F200D^4B10105D^3E2003^454F46
M^000007^05+COPY
M^000014^05+COPY
M^000027^05+COPY
E^000000
```

OUTPUT

```
Enter load address:- 3000
Location Object Code
3000 17202D
3003 69202D
3006 4B104036
300A 32026
300D 290000
3010 332007
3013 4B10405D
3017 3F3FEC
301A 32010
301C F2016
301F 10003
3022 F200D
3025 4B10405D
3029 3E2003
302C 454F46
```