Practical-8

Write a SAL program in text file and generate SYMTAB and LITTAB

```
#include<stdio.h>
#include<conio.h>
structsym
{
        char lab[10];
        intval;
};
void main ()
{
        FILE *f1,*fp;
        int i=0,j=0;
        char la[10],op[10],opr[10],a[1000],c,key[10],ch,d,l[10];
        inti,j,lc=0,m=0,flag,ch=0;
        structsym s[10];
        clrscr();
        f1=fopen("a1.txt","r");
        c=fgetc(f1);
        i=0;
        printf ("\n SOURCE PROGRAM \n");
        while(c!=EOF)
        {
                a[i]=c;
                c=fgetc(f1);
                i++;
        }
```

```
while(ch<4)
{
        printf("1-symbol table creation\n");
        printf("2-serch\n");
        printf("3-display\n");
        printf(">3-Exit\n");
        printf("enter ur choice\n");
        scanf("%d",&ch);
        switch(ch)
        {
                case 1:
                        i=0;
                        while(strcmp(op,"end")!=0)
                        {
                                if(a[i]=='\t')
                                {
                                        strcpy(la," ");
                                        i++;
                                }
                                else
                                {
                                        j=0;
                                        while(a[i] !='\t')
                                        {
                                                la[j]=a[i];
                                                i++;
                                                j++;
                                        }
                                        la[j]='\0';
        i++;
}
if(a[i]=='\t')
{
        strcpy(op," ");
        i++;
}
else
{
        j=0;
        while(a[i]!='\t')
        {
                op[j]=a[i];
                i++;
                j++;
        }
        op[j]='\0';
        i++;
}
if(a[i]=='\t')
{
        strcpy(opr," ");
        i++;
}
else
{
        j=0;
        while(a[i]!='\n')
        {
```

```
                        opr[j]=a[i];
                        i++;
                        j++;
                }
                opr[j]='\0';
                i++;
        }
        j=0;
        if(strcmp(la," ")!=0)
        {
                strcpy(s[m].lab,la);
                if(strcmp(op,"start")==0)
                {
                        lc=atoi(opr);
                        s[m].val=lc;
                        m++;
                        printf("%s\t%s\t%s\n",la,op,opr);
                        continue;
                }
                else if(strcmp(op,"equ")==0)
                {
                        s[m].val=atoi(opr);
                        m++;
                }
                else if(strcmp(op,"resw")==0)
                {
                        s[m].val=lc;
                        lc=lc+atoi(opr) *3;
                        m++;
```

```
                                }
                                else if(strcmp(op,"resb")==0)
                                {
                                        s[m].val=lc;
                                        lc=lc+atoi(opr);
                                        m++;
                                }
                                else
                                {
                                        s[m].val=lc;
                                        lc=lc+3;
                                        m++;
                                }
                        }
                        else
                                lc=lc+3;
                        printf("%s\t%s\t%s\n",la,op,opr);
                }
                break;
        case 2:
                printf("enter the lable to be searched\n");
                scanf("%s",&key);
                flag=0;
                for(i=0;i<m;i++)
                {
                        if(strcmp(key,s[i].lab)==0)
                        {
                                printf("%s\t%d\n",s[i].lab,s[i].val);
                                flag=1;
```

```
                                        break;
                                }
                                else
                                        continue;
                        }
                        if(flag==0)
                                printf("lable not found\n");
                                break;
                case 3:
                        printf("\n symbol table \n");
                        for(i=0;i<m;i++)
                                printf("\n%s\t%d\n",s[i].lab,s[i].val);
                                break;
                case 4:
                        printf("Literal table\n");
                        fp=fopen("sample.txt","r");
                        printf("literal table\n\nliteral value\t address\n");
                        do
                        {
                                ch=fgetc(fp);
                                if(ch=='=')
                                {
                                        ch=fgetc(fp);
                                        while(ch!=' ' &&ch!='\n' &&ch!=EOF &&ch!='\t')
                                        {
                                                l[j]=ch; printf( "%c",ch); ch=fgetc(fp);
                                                j++;
                                        }
                                        l[j]='\0';
                                        printf("\t\t%x",i+2);
```

```
                                                    printf("\n");
                                                    if((strlen(l))>=7)
                                                            i=i+4;
                                                    else
                                                            i=i+3;
                                                    j=0;
                                            }
                                    }while(ch!=EOF);
                                    fclose(fp);
                            }
                    }
            }
```

OUTPUT:

```
1-symbol table creation
2-search
3-display
4-Literal table
Enter your choice
3
Symbol table
Add 1000
Val 10
One 1009
Two 1012
1-symbol table creation
2-search
3-display
4-Literal table
```

```
Enter your choice
2
Enter the lable to be searched
Val
Val 10
1-symbol table creation
2-search
3-display
4-Literal table
Enter your choice
4
Literal table
Literal value address
x’057’ 2
k’0234’ 5
c’EOF’ 9
```