

**Lab 1: SPCC - C program for implementing Symbol Table****CODE**

```
# include <stdio.h>
# include <string.h>
# define null 0
int size=0;
void insert();
void del();
int search(char lab[]);
void modify();
void display();
struct symtab
{
    char label[10];
    int addr;
    struct symtab *next;
};
struct symtab *first,*last;
void main()
{
    int op;
    int y;
    char la[10];
    do
    {
        printf("\nSYMBOL TABLE IMPLEMENTATION\n");
        printf("1. INSERT\n");
        printf("2. DISPLAY\n");
        printf("3. DELETE\n");
        printf("4. SEARCH\n");
        printf("5. MODIFY\n");
        printf("6. END\n");
        printf("Enter your option : ");
        scanf("%d",&op);
        switch(op)
        {
            case 1:
                insert();
                display();
                break;
            case 2:
                display();
                break;
            case 3:
                del();
                display();
```

```

        break;
        case 4:
        printf("Enter the label to be searched : ");
        scanf("%s",la);
        y=search(la);
        if(y==1)
        {
        printf("The label is already in the symbol Table");
        }
        else
        {
        printf("The label is not found in the symbol table");
        }
        break;
        case 5:
        modify();
        display();
        break;
        case 6:
        break;
    }

}

while(op<6);
}

void insert()
{
    int n;
    char l[10];
    printf("Enter the label : ");
    scanf("%s",l);
    n=search(l);
    if(n==1)
    {
    printf("The label already exists. Duplicate cant be inserted\n");
    }
    else
    {
        struct symtab *p;
        p=malloc(sizeof(struct symtab));
        strcpy(p->label,l);
        printf("Enter the address : ");
        scanf("%d",&p->addr);
        p->next=null;
        if(size==0)
        {
            first=p;
            last=p;
        }
        else
        {

```

```

        last->next=p;
        last=p;
    }
    size++;
}

}

void display()
{
    int i;
    struct symtab *p;
    p=first;
    printf("LABEL\tADDRESS\n");
    for(i=0;i<size;i++)
    {
        printf("%s\t%d\n",p->label,p->addr);
        p=p->next;
    }
}

int search(char lab[])
{
    int i,flag=0;
    struct symtab *p;
    p=first;
    for(i=0;i<size;i++)
    {
        if(strcmp(p->label,lab)==0)
        {
            flag=1;
        }
        p=p->next;
    }
    return flag;
}

void modify()
{
    char l[10],nl[10];
    int add, choice, i, s;
    struct symtab *p;
    p=first;
    printf("What do you want to modify?\n");
    printf("1. Only the label\n");
    printf("2. Only the address of a particular label\n");
    printf("3. Both the label and address\n");
    printf("Enter your choice : ");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:
            printf("Enter the old label\n");

```

```

scanf("%s",l);
printf("Enter the new label\n");
scanf("%s",nl);
s=search(l);
if(s==0)
{
    printf("NO such label");
}
else
{
    for(i=0;i<size;i++)
    {
        if(strcmp(p->label,l)==0)
        {
            strcpy(p->label,nl);
        }
        p=p->next;
    }
}
break;
case 2:
printf("Enter the label whose address is to modified\n");
scanf("%s",l);
printf("Enter the new address\n");
scanf("%d",&add);
s=search(l);
if(s==0)
{
    printf("NO such label");
}
else
{
    for(i=0;i<size;i++)
    {
        if(strcmp(p->label,l)==0)
        {
            p->addr=add;
        }
        p=p->next;
    }
}
break;
case 3:
printf("Enter the old label : ");
scanf("%s",l);
printf("Enter the new label : ");
scanf("%s",nl);
printf("Enter the new address : ");
scanf("%d",&add);
s=search(l);
if(s==0)

```

```

        {
            printf("NO such label");
        }
        else
        {
            for(i=0;i<size;i++)
            {
                if(strcmp(p->label,l)==0)
                {
                    strcpy(p->label,l);
                    p->addr=add;
                }
                p=p->next;
            }
        }
        break;
    }
}

void del()
{
    int a;
    char l[10];
    struct symtab *p,*q;
    p=first;
    printf("Enter the label to be deleted\n");
    scanf("%s",l);
    a=search(l);
    if(a==0)
    {
        printf("Label not found\n");
    }
    else
    {
        {
            if(strcmp(first->label,l)==0)
            {
                first=first->next;
            }
            else if(strcmp(last->label,l)==0)
            {
                q=p->next;
                while(strcmp(q->label,l)!=0)
                {
                    p=p->next;
                    q=q->next;
                }
                p->next=null;
                last=p;
            }
        }
        else
        {
            q=p->next;

```

```

        while(strcmp(q->label,l)!=0)
        {
            p=p->next;
            q=q->next;
        }
        p->next=q->next;
    }
    size--;
}
}

```

## OUTPUT:

### SYMBOL TABLE IMPLEMENTATION

1. INSERT
2. DISPLAY
3. DELETE
4. SEARCH
5. MODIFY
6. END

Enter your option : 1

Enter the label : A

Enter the address : 10

LABEL ADDRESS

A 10

### SYMBOL TABLE IMPLEMENTATION

1. INSERT
2. DISPLAY
3. DELETE
4. SEARCH
5. MODIFY
6. END

Enter your option : 2

LABEL ADDRESS

A 10

### SYMBOL TABLE IMPLEMENTATION

1. INSERT
2. DISPLAY
3. DELETE
4. SEARCH
5. MODIFY
6. END

Enter your option : 1

Enter the label : B

Enter the address : 11

LABEL ADDRESS

A 10

B 11

#### SYMBOL TABLE IMPLEMENTATION

1. INSERT
2. DISPLAY
3. DELETE
4. SEARCH
5. MODIFY
6. END

Enter your option : 2

LABEL ADDRESS

A 10

B 11

#### SYMBOL TABLE IMPLEMENTATION

1. INSERT
2. DISPLAY
3. DELETE
4. SEARCH
5. MODIFY
6. END

Enter your option : 3

Enter the label to be deleted

B

LABEL ADDRESS

A 10

#### SYMBOL TABLE IMPLEMENTATION

1. INSERT
2. DISPLAY
3. DELETE
4. SEARCH
5. MODIFY
6. END

Enter your option : 2

LABEL ADDRESS

A 10

#### SYMBOL TABLE IMPLEMENTATION

1. INSERT
2. DISPLAY
3. DELETE
4. SEARCH
5. MODIFY
6. END

Enter your option : 1

Enter the label : C

Enter the address : 7

LABEL ADDRESS

A 10

C 7

#### SYMBOL TABLE IMPLEMENTATION

1. INSERT
2. DISPLAY
3. DELETE
4. SEARCH
5. MODIFY
6. END

Enter your option : 2

LABEL ADDRESS

A 10

C 7

#### SYMBOL TABLE IMPLEMENTATION

1. INSERT
2. DISPLAY
3. DELETE
4. SEARCH
5. MODIFY
6. END

Enter your option : 4

Enter the label to be searched : A

The label is already in the symbol Table

#### SYMBOL TABLE IMPLEMENTATION

1. INSERT
2. DISPLAY
3. DELETE
4. SEARCH
5. MODIFY
6. END

Enter your option : 5

What do you want to modify?

1. Only the label
2. Only the address of a particular label
3. Both the label and address

Enter your choice : 2

Enter the label whose address is to modified

A

Enter the new address

12

LABEL ADDRESS

A 12

C 7

#### SYMBOL TABLE IMPLEMENTATION

1. INSERT
2. DISPLAY
3. DELETE
4. SEARCH
5. MODIFY
6. END

Enter your option : 6