**2. insertion,deletion,search and modify in c language**

**# include <stdio.h>**

**# include <conio.h>**

**# include <alloc.h>**

**# include <string.h>**

**# define null 0**

**int size=0;**

**void insert();**

**void del();**

**int search(char lab[]);**

**void modify();**

**void display();**

**struct symbtab**

**{**

**char label[10];**

**int addr;**

**struct symtab \*next;**

**};**

**struct symbtab \*first,\*last;**

**void main()**

**{**

**int op;**

**int y;**

**char la[10];**

**clrscr();**

**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);**

**getch();**

**}**

**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 symbtab \*p;**

**p=malloc(sizeof(struct symbtab));**

**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 symbtab \*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 symbtab \*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 symbtab \*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,nl);**

**p->addr=add;**

**}**

**p=p->next;**

**}**

**}**

**break;**

**}**

**}**

**void del()**

**{**

**int a;**

**char l[10];**

**struct symbtab \*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--;**

**}**

**}**

**19.Well formed paranthesis**

**%{**

**/input is b.c/**

**#include<stdio.h>**

**#include<string.h>**

**char temp[10];**

**int i=0,openbracket=0,closebracket=0;**

**extern FILE \*yyin;**

**%}**

**%%**

**"("[()]\*")"";"{**

**strcpy(temp,yytext);**

**printf("%s\n",temp);**

**i=0;**

**while(temp[i]!=';')**

**{**

**if(temp[i]=='(')**

**openbracket++;**

**if(temp[i]==')')**

**closebracket++;**

**else ;**

**i++;**

**}**

**if(openbracket=closebracket)**

**printf("Well formed input!\n");**

**else**

**printf("not well formed!\n");**

**}**

**%%**

**main(int argc,char \*argv[])**

**{**

**FILE \*fp=fopen(argv[1],"r");**

**yyin=fp;**

**yylex();**

**fclose(fp);**

**}**

**18.To count the no. of lines,words,spaces and char of the given stmts**

**%{**

**#include<stdio.h>**

**int sc=0,wc=0,lc=0,cc=0;**

**%}**

**%%**

**[\n] { lc++; cc+=yyleng;}**

**[ \t] { sc++; cc+=yyleng;}**

**[^\t\n ]+ { wc++; cc+=yyleng;}**

**%%**

**int main(int argc ,char\* argv[ ])**

**{**

**printf("Enter the input:\n");**

**yylex();**

**printf("The number of lines=%d\n",lc);**

**printf("The number of spaces=%d\n",sc);**

**printf("The number of words=%d\n",wc);**

**printf("The number of characters are=%d\n",cc);**

**}**

**int yywrap( )**

**{**

**return 1;**

**}**

**14.Vowel consonants**

**%{**

**int vow\_count=0;**

**int const\_count =0;**

**%}**

**%%**

**[aeiouAEIOU] {vow\_count++;}**

**[a-zA-Z] {const\_count++;}**

**%%**

**int yywrap(){}**

**int main()**

**{**

**printf("Enter the string of vowels and consonants:");**

**yylex();**

**printf("Number of vowels are: %d\n", vow\_count);**

**printf("Number of consonants are: %d\n", const\_count);**

**return 0;**

**}**

**11.Generate a valid pattern that recognizes all statements that begins with an Upper-Case Letter followed by five digits or alphabets. Use a YACC tool to do the same.**

**/\* Definition Section \*/**

**%{**

**int valid = 0;**

**int invalid = 0;**

**%}**

**/\* Rules Section \*/**

**%%**

**([A-Z][a-zA-Z0-9][a-z]) {printf("\n\tPattern Detected: %s ", yytext); valid++;}**

**[a-zA-Z0-9]\* {invalid++;}**

**"\n" {**

**printf("\n\n\tNumber of VALID patterns = %d\n", valid);**

**printf("\tNumber of invalid patterns = %d\n\n", invalid);**

**valid = 0;**

**invalid = 0;**

**}**

**EXIT\_\_ return 0;**

**%%**

**/\* User code section\*\*\*/**

**int yywrap(){}**

**int main(int argc, char \*\*argv[])**

**{**

**printf("\n Enter inputs: \n\n");**

**yylex();**

**return 0;**

**}**

**16.Reverse scan words**

**%{**

**#include <stdio.h>**

**#include <string.h>**

**%}**

**/\* Regular expressions for tokens \*/**

**%%**

**"auto"|"break"|"case"|"char"|"const"|"continue"|"default"|"do"|"double"|"else"|"enum"|"extern"|"float"|"for"|"goto"|"if"|"int"|"long"|"register"|"return"|"short"|"signed"|"sizeof"|"static"|"struct"|"switch"|"typedef"|"union"|"unsigned"|"void"|"volatile"|"while" {**

**printf("Reserved Word: %s\n", yytext);**

**}**

**[\_a-zA-Z][\_a-zA-Z0-9]\* {**

**printf("Variable: %s\n", yytext);**

**}**

**"+"|"-"|"\*"|"/"|"="|"=="|"!="|"<"|">"|"<="|">="|"++"|"--"|"&&"|"||"|"!"|"&"|"|"|"%" {**

**printf("Operator: %s\n", yytext);**

**}**

**[ \t\n] ;**

**. {**

**printf("Invalid Character: %s\n", yytext);**

**}**

**%%**

**int main() {**

**printf("Enter C code (press Ctrl+D to end):\n");**

**yylex();**

**return 0;**

**}**

**int yywrap(){**

**return 1;**

**}**