//shift-reduce parser

//The chosen grammar is

//(1)E' → E

//(2)E → aEa

//(3)E → b

//The corresponding parsing table is

//      a       b       $   |   E

//0     s2      s3          |   1

//1                     ac  |

//2     s2      s3          |   4

//3     r3              r3  |

//4     s5                  |

//5     r2              r2  |

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int i, c, flag=1,k=0;

char a[16], stk[15];

void gotofunc(){

    if(stk[k]=='E' && stk[k-1]=='0'){

        stk[++k] = '1';

    }

    else if(stk[k]=='E' && stk[k-1]=='2'){

        stk[++k] = '4';

    }

    else{

        printf("Error in function!");

        exit(0);

    }

}

void take\_action(int c){

    printf("\n%s\t\t%s\t\t", stk, a);

    //printf("%d\_%d\n", k, c);

    if(a[c] == '$' && stk[k]=='1'){

        flag = 1;

        printf("ACCEPT\n");

        return;

    }

    else if(a[c] == 'a' && stk[k]=='2'){

        //s2

        stk[++k] = a[c];

        stk[++k] = '2';

        a[c] = ' ';

        printf("SHIFT\n");

    }

    else if(a[c] == 'a' && stk[k]=='0'){

        //s2

        stk[++k] = a[c];

        stk[++k] = '2';

        a[c] = ' ';

        printf("SHIFT\n");

    }

    else if(a[c] == 'b' && stk[k]=='0'){

        //s3

        stk[++k] = a[c];

        stk[++k] = '3';

        a[c] = ' ';

        printf("SHIFT\n");

    }

    else if(a[c] == 'b' && stk[k]=='2'){

        //s3

        stk[++k] = a[c];

        stk[++k] = '3';

        a[c] = ' ';

        printf("SHIFT\n");

    }

    else if(a[c] == 'a' && stk[k]=='3'){

        //r3

        stk[k]=' ';

        k = k-1;

        stk[k] = 'E';

        printf("REDUCE E --> b\n");

        gotofunc();

        take\_action(c);

    }

    else if(a[c] == '$' && stk[k]=='3'){

        //r3

        stk[k]=' ';

        k = k-1;

        stk[k] = 'E';

        printf("REDUCE E --> b\n");

        gotofunc();

        take\_action(c);

    }

    else if(a[c] == 'a' && stk[k]=='4'){

        //s5

        stk[++k] = a[c];

        stk[++k] = '5';

        a[c] = ' ';

        printf("SHIFT\n");

    }

    else if(a[c] == 'a' && stk[k]=='5'){

        //r2

        for(int l=0;l<5;l++){

            stk[k] = ' ';

            k--;

        }

        stk[k] = 'E';

        printf("REDUCE E --> aEa\n");

        gotofunc();

        take\_action(c);

    }

    else if(a[c] == '$' && stk[k]=='5'){

        //r2

        for(int l=0;l<5;l++){

            stk[k] = ' ';

            k--;

        }

        stk[k] = 'E';

        printf("REDUCE E --> aEa\n");

        gotofunc();

        take\_action(c);

    }

    else{

        printf("ERROR!\n");

        flag = 0;

        return;

    }

}

int main()

{

    char\* in\_str;

    printf("GRAMMAR is -\nE->aEa \nE->b\n");

    printf("Enter input string:\t");

    scanf("%s", in\_str);

    strcpy(a,strcat(in\_str,"$"));

    c=strlen(a);

    stk[k] = '0';

    printf("\nstack \t\t input \t\t action");

    for(i = 0; i < c; i++){

        // print the values of stack and input

        take\_action(i);

        if(flag==0){

            break;

        }

    }

    // if top of the stack is E(starting symbol), then it will accept the input

    if(flag == 1){

        printf("String accepted successfully\n");

    }

    else{

        printf("String rejected\n");

    }

}