```
program sp;

{------------------------Mehdi Mohammadi sn. : 860934685------------------------}

uses crt;
const
    m=20;
type stack=record
    items                                                                   :array[1..m] of char;
    mytop                                                                   :integer;
    end;
type
    t1array=array [1..10,1..10] of char;
    t2array=array [1..10] of string;

{--------------------------------------------------------------------------------}

function find_index(temp_char_1:char;col:integer;table:t1array):integer;
var
    i                                                                       :integer;
begin
    for i:=1 to col do
        begin
            if table[1,i] = temp_char_1 then
                begin
                    find_index:=i;
                end
        end;
end;

{--------------------------------------------------------------------------------}

function empty(s:stack):boolean;
begin
    if s.mytop = 0 then
        empty := true
    else
        empty := false
end;

{--------------------------------------------------------------------------------}

procedure popandtest(var s:stack; var x:char; var underflow:integer);
begin
    if empty(s) then
        underflow := 1
    else
        begin
            x := s.items[s.mytop];
            s.mytop := s.mytop - 1;
            underflow := 0;
        end;
end;

{--------------------------------------------------------------------------------}

procedure topandtest(var s:stack; var x:char; var underflow:integer);
begin
    if empty(s) then
        underflow := 1
    else
        begin
            x := s.items[s.mytop];
            underflow := 0;
        end;
end;

{--------------------------------------------------------------------------------}

procedure pushandtest(var s:stack; x:char; var underflow:integer);
begin
    if s.mytop = m then
        underflow := 1
    else
        begin
            s.mytop := s.mytop + 1;
            s.items[s.mytop] := x;
            underflow := 0;
        end;
end;

{--------------------------------------------------------------------------------}

```

```
 84  procedure print(s:stack);
 85  var
 86      i                                                                   :integer;
 87      output_string                                                       :string;
 88  begin
 89      output_string:='';
 90      if empty(s) then
 91          write('Stack Is Empty. Press A Key');
 92      for i := 1 to s.mytop do
 93          begin
 94              output_string:=output_string+s.items[i];
 95          end;
 96      write(output_string:12);
 97  end;
 98
 99  {-------------------------------------------------------------------------------}
100
101  function compare(stack_char:char;input_char:char;col:integer;table:t1array):char;
102  begin
103      compare:=table[find_index(stack_char,col,table),find_index(input_char,col,table)];
104  end;
105
106  {-------------------------------------------------------------------------------}
107
108  var
109      table_file,grammer_file                                             :text;
110      s                                                                   :stack;
111      stack_char,check_status,ch                                          :char;
112      check_stack,row,col,i,j                                             :integer;
113      flag                                                                :boolean;
114      input,temp_string_1,temp_string_2,test,output_string                :string;
115      table                                                               :t1array;
116      grammer                                                             :t2array;
117  begin
118      clrscr;
119      s.mytop:=0;
120      output_string:='';
121      row:=1;
122      col:=1;
123      assign(table_file,'sp.txt');
124      reset(table_file);
125      while not eof(table_file) do
126          begin
127              if row=1 then
128                  begin
129                      while not eoln(table_file) do
130                          begin
131                              read(table_file,ch);
132                              if ch <> ' ' then
133                                  begin
134                                      table[row,col]:=ch;
135                                  end
136                              else
137                                  begin
138                                      col:=col+1;
139                                  end;
140                          end;
141                      row:=row+1;
142                  end
143              else
144                  begin
145                      col:=1;
146                      while not eoln(table_file) do
147                          begin
148                              read(table_file,ch);
149                              if ch <> ' ' then
150                                  begin
151                                      table[row,col]:=ch;
152                                  end
153                              else
154                                  begin
155                                      col:=col+1;
156                                  end;
157                          end;
158                      row:=row+1;
159                  end;
160              readln(table_file);
161          end;
162      close(table_file);
163
164  {-------------------------------------------------------------------------------}
165
166      assign(grammer_file,'grammer.txt');
```

```
167       reset(grammer_file);
168       row:=1;
169       while not eof(grammer_file) do
170           begin
171               while not eoln(grammer_file) do
172                   begin
173                       read(grammer_file,ch);
174                       grammer[row]:=grammer[row]+ch;
175                   end;
176               row:=row+1;
177               readln(grammer_file);
178           end;
179       close(grammer_file);
180
181   {--------------------------------------------------------------------------------}
182
183                               {>>>Display Input Data<<<}
184       col:=col-1;
185       row:=row-1;
186       textcolor(lightmagenta);
187       for i:=1 to col do
188           begin
189               for j:=1 to col do
190                   begin
191                       if (i=1) or (j=1) then
192                           textcolor(lightmagenta)
193                       else
194                           textcolor(yellow);
195                       write(table[i,j]);
196                   end;
197               writeln;
198            end;
199       writeln;
200       for i:=1 to row do
201           begin
202               textcolor(lightmagenta);
203               write(i,' : ');
204               textcolor(yellow);
205               write(grammer[i]);
206               writeln;
207           end;
208       writeln;
209       textcolor(lightcyan);
210                               {>>>Display Input Data<<<}
211
212   {--------------------------------------------------------------------------------}
213
214       flag:=false;
215       while not flag do
216           begin
217               write('Please Insert Your String : ');
218               readln(input);
219               for i:=1 to length(input) do
220                   begin
221                       for j:=2 to col do
222                           begin
223                               if (table[1,j]=input[i]) and (input[i]<>'$') then
224                                   begin
225                                       flag:=true;
226                                       break;
227                                   end
228                               else
229                                   begin
230                                       flag:=false;
231                                   end;
232                           end;
233                       for j:=1 to row do
234                           begin
235                               if grammer[j,1] = input[i] then
236                                   begin
237                                       flag:=false;
238                                       break;
239                                   end;
240                           end;
241                       if flag=false then
242                           begin
243                               writeln;
244                               writeln('Incorrect String !');
245                               writeln;
246                               break;
247                           end;
248                   end;
249           end;
```

```
250      writeln;
251      input:=input+'$';
252
253  {--------------------------------------------------------------------------------}
254
255      pushandtest(s,'$',check_stack);
256      i:=1;
257      print(s);
258      output_string:='';
259      for j:=i to length(input)do
260          begin
261              output_string:=output_string+input[j];
262          end;
263      write(output_string:12);
264      while i <= length(input) do
265          begin
266              topandtest(s,stack_char,check_stack);
267              check_status:=compare(stack_char,input[i],col,table);
268
269  {--------------------------------------------------------------------------------}
270
271              if check_status='E' then
272                  begin
273                      writeln('Syntax Error !':23);
274                      writeln;
275                      write('Press Any Key To Exit...');
276                      readln;
277                      exit;
278                  end
279
280  {--------------------------------------------------------------------------------}
281
282              else if check_status='=' then
283                  begin
284                      write('Shift ':15);
285                      writeln;
286                      pushandtest(s,input[i],check_stack);
287                      i:=i+1;
288                      print(s);
289                      output_string:='';
290                      for j:=i to length(input)do
291                          begin
292                              output_string:=output_string+input[j];
293                          end;
294                      write(output_string:12);
295                  end
296
297  {--------------------------------------------------------------------------------}
298
299              else if check_status='<' then
300                  begin
301                      write('Shift ':15);
302                      writeln;
303                      pushandtest(s,'<',check_stack);
304                      pushandtest(s,input[i],check_stack);
305                      i:=i+1;
306                      print(s);
307                      output_string:='';
308                      for j:=i to length(input)do
309                          begin
310                              output_string:=output_string+input[j];
311                          end;
312                      write(output_string:12);
313                  end
314
315  {--------------------------------------------------------------------------------}
316
317              else if check_status='>' then
318                  begin
319                      temp_string_1:='';
320                      temp_string_2:='';
321                      while true do
322                          begin
323                              popandtest(s,stack_char,check_stack);
324                              if check_stack = 1 then
325                                  begin
326                                      writeln('Syntax Error !':23);
327                                      writeln;
328                                      write('Press Any Key To Exit...');
329                                      readln;
330                                      exit;
331                                  end;
332                              if stack_char='<' then
```

```
                                    begin
                                        write('Reduce':15);
                                        break;
                                    end;
                                temp_string_1:=temp_string_1+stack_char;
                            end;
                        for j:=length(temp_string_1) downto 1 do
                            temp_string_2:=temp_string_2+temp_string_1[j];
                        temp_string_1:='';
                        for j:=1 to row do
                            begin
                                temp_string_1:=copy(grammer[j],4,length(grammer[j])-3);
                                if temp_string_1=temp_string_2 then
                                    break;
                            end;
                        write('     ',j,' : ',grammer[j]);
                        writeln;
                        topandtest(s,stack_char,check_stack);
                        check_status:=compare(stack_char,grammer[j,1],col,table);
                        if check_status='<' then
                            begin
                                pushandtest(s,'<',check_stack);
                                pushandtest(s,grammer[j,1],check_stack);
                                print(s);
                                output_string:='';
                                for j:=i to length(input)do
                                    begin
                                        output_string:=output_string+input[j];
                                    end;
                                write(output_string:12);
                            end;
                        if check_status='=' then
                            begin
                                pushandtest(s,grammer[j,1],check_stack);
                                print(s);
                                output_string:='';
                                for j:=i to length(input)do
                                    begin
                                        output_string:=output_string+input[j];
                                    end;
                                write(output_string:12);
                            end;
                    end

{--------------------------------------------------------------------------------}

            end;
        for i:=1 to 3 do
            begin
                popandtest(s,stack_char,check_stack);
                test:=test+stack_char;
            end;
        if test='$s$' then
            write('Accept':15)
        else
            write('Not Accept':19);
        writeln;
        writeln;
        write('Press Any Key To Exit...');
        readln;
    end.
```