Remal Public School
A-2, Sector-3, Rohini, Delhi - 110085

Test Your Knowledge!

A C++ project

Author:
Arpit Saxena
XI-E
Roll No. 6

February 16, 2017

Contents

Source Code

Header files

1. quiz.h

```
/*!
 \file quiz.h
 \brief Contains prototypes of all functions required
*/

#ifndef QUIZ_H
#define QUIZ_H

#include "helpers.h"
#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
#include <time.h>
#include <stdio.h>
#include <ctype.h>
#include <string.h>


//! Structure to represent a question
struct question
{
    char q[200]; //!< The question
    char options[4][100]; //!< The options
    int correct; //!< The correct option (option count starts from 1)
};

//ques.cpp

//! Initialises the struct questions
/*!
 \sa question
 @return void
*/
void init_ques();

//ui.cpp

//! Initialises the User Interface by setting general requirements for other UI functions
/*!
 @return void
*/
void init_ui();

//! Prints the input string at center of line
/*!
 Prints the input string at center; If ws is defined, it prints the string with ws
 number of ' ' characters before the string. The text color and background color are
 specified through t_color and b_color respectively.
```

```
 @param[in] str     The input string to be printed
 @param[in] ws      Amount of whitespace from left end (-1, if not specified)
 @param[in] t_color Text color (LIGHTGRAY, if not specified)
 @param[in] b_color Background color (BLACK, if not specified)
 @return void
*/
void printc(char* str, int ws = (-1), int t_color = LIGHTGRAY, int b_color =
    BLACK);

//! Structure to represent a coordinate of type (x,y)
struct coord
{
    //! Constructor with inputs
    /*!
     @param[in] x_inp The x-coordinate to be set
     @param[in] y_inp The y-coordinate to be set
    */
    coord(int x_inp, int y_inp)
    {
        x = x_inp;
        y = y_inp;
    }

    //! Default constructor
    /*!
     Sets both x and y coordinates to 0
    */
    coord()
    {
        x = 0;
        y = 0;
    }
    int x; //!< x coordinate
    int y; //!< y coordinate
};


//! Generates a frame
/*!
 @param[in] up_left  Upper left coordinate of frame; upper-left corner of window,
      if not specified
 @param[in] f_height Height of frame
 @param[in] f_width  Width of frame
 @param[in] f_sides  Specifies whether sides of frame should be printed (1 = true
     , 0 = false)
 @return void
 \sa coord
*/
void frame (coord up_left = coord(-1,-1), int f_height = -1, int f_width = -1,
    int f_sides = 1);

//! Generates UI for specified screen
/*!
 The UI generated includes selection of option and putting them in a frame
 @param[in] screen_num Screen Number (1: Difficulty, 2: Subject)
 @return The line selected (starting from 0)
 \sa frame(), printc(), coord, select()
*/
int generate_ui(int screen_num);
```

```
104
105 //! Generates UI for question screen
106 /*!
107  The UI generated includes selection of option and putting them in a frame
108  @param[in] lvl   Difficulty Level
109  @param[in] sub   Subject
110  @param[in] q_num Question Number (starts from 0)
111  @return The option selected (starting from 0)
112  \sa frame(), printc(), coord, select()
113 */
114 int generate_ui(int lvl, int sub, int q_num);
115
116 //askq.cpp
117
118 //! Asks questions
119 /*!
120  Uses generate_ui() to ask questions, keeps track of correct answers, prints
121  it at the end of quiz and asks the user whether to continue playing.
122  @param[in] lvl   Difficulty Level
123  @param[in] sub   Subject
124  @return Whether user wants to play again (0: no, 1: yes)
125  \sa generate_ui(), frame(), printc()
126 */
127 int ask_q(int lvl, int sub);
128
129 //! Makes the user select the option
130 /*!
131  Takes in coordinates of first bullet, number of options and prints the bullets
132  according to the height of each option. Also, takes in the character to be
       printed
133  as a bullet
134  @param[in] bullet1    Co-ordinate of first bullet
135  @param[in] num_ops    Number of options
136  @param[in] height_ops Height of each option
137  @param[in] bullet     Character to be printed as bullet
138  @return The selected option (starts from 0)
139  \sa coord
140 */
141 int select(coord bullet1, int num_ops, int* height_ops, char bullet = (char) 175)
       ;
142
143 /*!
144  \overload
145  @param[in] bullet1    Co-ordinate of first bullet
146  @param[in] num_ops    Number of options
147  @param[in] bullet     Character to be printed as bullet
148  @return The selected option (starts from 0)
149 */
150 int select(coord bullet1, int num_ops, char bullet = (char) 175);
151
152
153 #endif //QUIZ_H
```

2. helpers.h

```
/*!
 \file helpers.h
 \brief Contains helper functions such as wrap
*/

#ifndef HELPERS_H
#define HELPERS_H

#include <stdio.h>
#include <string.h>

//! Wraps a string with specified number of characters in each line
/*!
 Takes in an input string and changes the output string with length (or less)
 number of characters, each separated by '\n'
 @param[in]  inp_str The input string
 @param[out] out_str The output string
 @param[in]  length  The max. number of characters in each line
 @return The number of lines contained in out_str
*/
int wrap(char* inp_str, char* out_str, int length);

#endif //HELPERS_H
```

C++ files (.cpp)

1. main.cpp : Contains main function and calls other functions

```cpp
#include "quiz.h"

extern question questions[3][3][3];

void main()
{
    clrscr();

    //! Boolean value indicating if user wants to play again
    int play = 1;

    while(play)
    {
        init_ques();
        init_ui();

        int lvl = generate_ui(1);

        int sub = generate_ui(2);

        play = ask_q(lvl, sub);
    }
}
```

2. askq.cpp : Calls the UI functions and manages number of correct answers given by user

```
#include "quiz.h"

extern question questions[3][3][3];
extern height;
extern width;

int ask_q(int lvl, int sub)
{
    int correct_ans = 0; //!< Number of questions answered correctly
    for(int i = 0; i < 3; i++)
    {
        int is_correct = generate_ui(lvl, sub, i); //!< Determines if the option
            chosen by the user is correct or not

        if(is_correct)
        {
            correct_ans++;
        }
        getch();
    }

    clrscr();

    while(1)
    {
        frame();
        gotoxy((width - strlen("You got _ out of 3 questions correct")) / 2 + 1,
            (height - 3) / 2 + 1);
        cout << "You got " << correct_ans << " out of 3 questions correct\n\n";
        printc("Do you want to play again? (y/n)");
        char response = getch(); //!< Response from the user whether to play
            again or not
        switch(c)
        {
            case 'y':
            case 'Y':
                return 1;
            case 'n':
            case 'N':
                return 0;
        }
    }
}
```

3. ui.cpp : Contains UI functions that generate the UI and interact with the user

```
#include "quiz.h"

#define TITLE "TEST YOUR KNOWLEDGE!" //!< Title for the quiz

extern question questions[3][3][3];

int height; //!< Height of the screen
int width; //!< Width of the screen

void init_ui()
{
    _setcursortype(_NOCURSOR); //!<hides cursor
    textcolor(LIGHTGRAY);
    textbackground(BLACK);

    struct text_info info; //!< Information regarding the screen, etc.
    gettextinfo(&info);

    //height and witdth of screen
    width = (int) info.screenwidth;
    height = (int) info.screenheight;
}

void frame(coord up_left, int f_height, int f_width, int f_sides)
{
    //! Characters for printing the frame
    char l_up = (char) 201;
    char l_down = (char) 200;
    char r_up = (char) 187;
    char r_down = (char) 188;
    char bet_ver = (char) 186;
    char bet_hor = (char) 205;

    if(up_left.x == -1)
    {
        coord b(1,1); //! Upper left corner of the screen
        up_left = b;
        f_width = width;
        f_height = height - 1;
    }

    gotoxy(up_left.x, up_left.y);

    cout << l_up;
    for(int i = 0; i < f_width - 2; i++)
    {
        cout << bet_hor;
    }
    cout << r_up;

    if(f_sides)
    {
        for(i = 2; i <= f_height - 1; i++)
        {
            gotoxy(up_left.x, i); cout << bet_ver;
            gotoxy(up_left.x + f_width - 1, i); cout << bet_ver;
```

```
57          }
58      }
59
60      gotoxy(up_left.x, up_left.y + f_height + 1);
61      cout << l_down;
62      for(i = 0; i < f_width - 2; i++)
63      {
64          cout << bet_hor;
65      }
66      cout << r_down;
67
68      gotoxy(up_left.x + 2, up_left.y + 2);
69  }
70
71  void printc(char str[], int ws, int t_color, int b_color)
72  {
73      if(ws == -1)
74      {
75          gotoxy((width - strlen(str))/2 + 1 ,wherey());
76      }
77      else
78      {
79          gotoxy(ws + 1, wherey());
80      }
81
82      textcolor(t_color);
83      textbackground(b_color);
84      cprintf("%s", str);
85  }
86
87  /*
88   Generates user interface
89   Screen_num:
90      1: difficulty level
91      2: Subject
92  */
93  int generate_ui(int screen_num)
94  {
95      clrscr();
96      frame();
97
98      char title[] = TITLE; //!< title for the quiz
99      char head[100]; //!< head of the section
100
101     char op1[100], op2[100], op3[100];
102
103     if(screen_num == 1)
104     {
105         strcpy(head, "Select difficulty level: ");
106         strcpy(op1, "Easy");
107         strcpy(op2, "Intermediate");
108         strcpy(op3, "Hard");
109     }
110     else if(screen_num == 2)
111     {
112         strcpy(head, "Select subject: ");
113         strcpy(op1, "Computer Science (C++)");
114         strcpy(op2, "General Knowledge");
115         strcpy(op3, "English");
```

```
116     }
117
118     gotoxy(1, (height - 6) / 2);
119
120     printc(title);
121     cout << "\n\n";
122
123     int lengths[] = {
124                       strlen(head),
125                       strlen(op1),
126                       strlen(op2),
127                       strlen(op3),
128                     }; //!< string lengths of head and options
129
130     //Determining max length
131     int max_len = lengths[0]; //!< The maximum value of string lengths stored in
            lengths
132     for(int i = 0; i < 4; i++)
133     {
134         for(int j = i + 1; j < 4; j++)
135         {
136             if(lengths[j] > max_len)
137             {
138                 max_len = lengths[j];
139             }
140         }
141     }
142
143     int a = (width - max_len) / 2; //!< Whitespace before head and all options
144     coord b(a - 4,wherey()); //!< Upper left corner of internal frame
145     frame(b, 6, max_len + 8, 0);
146
147     printc(head, a);
148     cout << '\n';
149
150     coord bullet1(b.x + 2, wherey()); //!< Position of the first bullet
151
152     gotoxy(bullet1.x + 2, bullet1.y); printc(op1, a); cout << '\n';
153     gotoxy(bullet1.x + 2, bullet1.y + 1); printc(op2, a); cout << '\n';
154     gotoxy(bullet1.x + 2, bullet1.y + 2); printc(op3, a); cout << '\n';
155
156     return select(bullet1, 3);
157 }
158
159 int generate_ui(int lvl, int sub, int q_num)
160 {
161     clrscr();
162     frame();
163
164     gotoxy(2,2); cout << "Difficulty: ";
165     switch(lvl)
166     {
167         case 0:
168             cout << "Easy";
169             break;
170         case 1:
171             cout << "Intermediate";
172             break;
173         case 2:
```

```
174             cout << "Hard";
175             break;
176     }
177
178     char print[100] = "Subject: ";
179     switch(sub)
180     {
181         case 0:
182             strcat(print, "C++");
183             break;
184         case 1:
185             strcat(print, "G.K.");
186             break;
187         case 2:
188             strcat(print, "English");
189             break;
190     }
191     gotoxy(width - strlen(print), 2);
192     cout << print;
193
194     char ques[200]; //!< String containing questions
195     char options[4][200]; //!< String array containing the four options
196
197     strcpy(ques, questions[lvl][sub][q_num].q);
198     strcpy(options[0], questions[lvl][sub][q_num].options[0]);
199     strcpy(options[1], questions[lvl][sub][q_num].options[1]);
200     strcpy(options[2], questions[lvl][sub][q_num].options[2]);
201     strcpy(options[3], questions[lvl][sub][q_num].options[3]);
202
203     int f_width = (8 * width) / 10; //!< Width of the internal frame
204     int ws = (width - f_width) / 2; //!< Whitespace before the left edge of frame
205
206     int lengths[] = {
207                      strlen(ques),
208                      strlen(options[0]),
209                      strlen(options[1]),
210                      strlen(options[2]),
211                      strlen(options[3])
212                     }; //!< array that stores lengths of the question and options
213
214     int f_height = 2 + lengths[0] / f_width + 1; //!< Height of the internal
            frame
215     int f_ws = 3;  //!< Whitespace inside frame for options
216     for(int i = 1; i < 5; i++)
217     {
218         f_height += lengths[i] / (f_width - f_ws) + 1;
219     }
220
221     coord u_left(ws + 1, (height - f_height) / 2 + 1);
222     frame(u_left, f_height, f_width, 0);
223
224     char printq[] = "Question "; //!< The question string to be printed
225     gotoxy((width - strlen(printq) - 1) / 2 + 1, 2);
226     cout << printq << q_num + 1;
227
228     gotoxy(1, (height - f_height) / 2 - 3);
229     printc(TITLE);
230
231     char fstring[200] = ""; //!< Formatted string for out_str of wrap()
```

```
232
233     int height_ops[4] = {0,0,0,0};
234     int height_ques = wrap(ques, fstring, f_width - f_ws);
235
236     int line_num = 1;
237     char line[200];
238     int read; //!< Number of characters read in current line
239     int chars_read = 0; //!< Total number of characters read
240
241     //! Coordinates from where to start printing question
242     coord ques_coord(u_left.x + 1, u_left.y + 2);
243
244     for(i = 0; i < height_ques; i++)
245     {
246         sscanf(fstring + chars_read, "%[^\n]%*c%n", line, &read);
247         gotoxy(ques_coord.x + 2, ques_coord.y + line_num - 1);
248         cout << line;
249         line_num++;
250         chars_read += read;
251     }
252
253     strcpy(fstring, "");
254
255     //! Coordinates of first bullet
256     coord bullet1(ques_coord.x, ques_coord.y + height_ques);
257
258     for(i = 0; i < 4; i++)
259     {
260         height_ops[i] = wrap(options[i], fstring, f_width - f_ws - 3 - 2);
261         read = 0, chars_read = 0;
262
263         int len = strlen(fstring); //!< length of fstring
264         fstring[len] = '\n'; fstring[len+1] = '\0';
265
266         //! x: +2 space for bullet, +3 space for option letter
267         coord op_start(bullet1.x + 2 + 3, bullet1.y);
268
269         for(int j = 0; j < i; j++)
270         {
271             op_start.y += height_ops[j];
272         }
273
274         gotoxy(bullet1.x + 2, op_start.y);
275         cout << (char) ('A' + i) << ". ";
276
277         for(j = 0; j < height_ops[i]; j++)
278         {
279             sscanf(fstring + chars_read, "%[^\n]%*c%n", line, &read);
280             gotoxy(op_start.x, op_start.y + j);
281             line_num++;
282             cout << line;
283             chars_read += read;
284         }
285
286         strcpy(fstring, "");
287     }
288
289     //! The option selected by the user (counts from 1)
290     int selected_ans = select(bullet1, 4, height_ops) + 1;
```

```
291     gotoxy(1, (height - f_height) / 2 - 1);
292     if(selected_ans == questions[lvl][sub][q_num].correct)
293     {
294         printc("Correct Answer!!");
295         return 1;
296     }
297     else
298     {
299         char pr[] = "Incorrect Answer!! Correct answer was ";
300         char a[2] = {(char) ('A' + questions[lvl][sub][q_num].correct - 1),'\0'};
301         strcat(pr,a);
302         printc(pr);
303         return 0;
304     }
305 }
306
307 int select(coord bullet1, int num_ops, char bullet)
308 {
309     int height_ops[] = {1,1,1,1,1};
310     return select(bullet1, num_ops, height_ops, bullet);
311 }
312
313 int select(coord bullet1, int num_ops, int height_ops[], char bullet)
314 {
315     //! current line on which the bullet is on (counts from 0)
316     int curr_line = 0;
317     gotoxy(bullet1.x, bullet1.y); cout << bullet;
318     while(1)
319     {
320         char c = getch(); //!< Input from user
321
322         if(c == 0)
323         {
324             c = getch();
325             switch(c)
326             {
327                 case 'H':
328                     curr_line == 0 ? curr_line = num_ops - 1 : curr_line--;
329                     break;
330                 case 'P':
331                     curr_line == num_ops - 1 ? curr_line = 0 : curr_line++;
332                     break;
333             }
334         }
335         else if(c == 13)
336         {
337             return curr_line;
338         }
339
340         for(int i = 0; i < num_ops; i++)
341         {
342             coord bullet_pos;
343
344             bullet_pos.x = bullet1.x;
345             bullet_pos.y = bullet1.y;
346             for(int j = 0; j < i; j++)
347             {
348                 bullet_pos.y += height_ops[j];
349             }
```

```
350
351             gotoxy(bullet_pos.x, bullet_pos.y);
352
353             if(i == curr_line)
354             {
355                 cout << bullet;
356             }
357             else
358             {
359                 cout << ' ';
360             }
361         }
362     }
363 }
```

4. helpers.cpp : Contains helper functions such as wrap

```
#include "helpers.h"

int wrap(char inp_str[], char out_str[],int length)
{
    char word[30];  //!< word extracted from inp_str

    int chars_read = 0; //!< total number of chararacter read from inp_str
    int read; //!< number of characters read for the current line
    int num_lines = 1; //!< Number of lines written into out_str

    while(sscanf(inp_str + chars_read, "%s%n", word, &read) > 0)
    {
        if(written + strlen(word) > length)
        {
            strcat(out_str, "\n");
            num_lines++;
            strcat(out_str, word);
            strcat(out_str, " ");
        }
        else
        {
            strcat(out_str, word);
            strcat(out_str, " ");
        }
        chars_read += read;
    }

    return num_lines;
}
```

5. ques.cpp : Contains the questions (taken from the .txt files) arranged in a structure

```
/* Generated from script.py */
#include "quiz.h"
#include <string.h>

question questions[3][3][3]; //!< 3d array to store questions - questions[num_levels][num_subjects][num_qs]

void init_ques()
{

strcpy(questions[0][0][0].q, "Select the INCORRECT keyword in C++") ;
strcpy(questions[0][0][0].options[0], "asm") ;
strcpy(questions[0][0][0].options[1], "virtual") ;
strcpy(questions[0][0][0].options[2], "statics") ;
strcpy(questions[0][0][0].options[3], "float") ;
questions[0][0][0].correct = 3 ;

strcpy(questions[0][0][1].q, "String literal \"abc\" will be represented in the memory as _____") ;
strcpy(questions[0][0][1].options[0], "abc\\0") ;
strcpy(questions[0][0][1].options[1], "abc/0") ;
strcpy(questions[0][0][1].options[2], "abc|0") ;
strcpy(questions[0][0][1].options[3], "abc_0") ;
questions[0][0][1].correct = 1 ;

strcpy(questions[0][0][2].q, "Which of the following header files contains the exit () function to terminate the current program in C++?") ;
strcpy(questions[0][0][2].options[0], "<string.h>") ;
strcpy(questions[0][0][2].options[1], "<process.h>") ;
strcpy(questions[0][0][2].options[2], "<iomanip.h>") ;
strcpy(questions[0][0][2].options[3], "<ctype.h>") ;
questions[0][0][2].correct = 2 ;

strcpy(questions[0][1][0].q, "Entomology is the science that studies ______") ;
strcpy(questions[0][1][0].options[0], "Behavior of human beings") ;
strcpy(questions[0][1][0].options[1], "Insects") ;
strcpy(questions[0][1][0].options[2], "The origin and history of technical and scientific terms") ;
strcpy(questions[0][1][0].options[3], "The formation of rocks") ;
questions[0][1][0].correct = 2 ;

strcpy(questions[0][1][1].q, "Hitler party which came into power in 1933 is known as") ;
strcpy(questions[0][1][1].options[0], "Labour Party") ;
strcpy(questions[0][1][1].options[1], "Nazi Party") ;
strcpy(questions[0][1][1].options[2], "Ku-Klux-Klan") ;
strcpy(questions[0][1][1].options[3], "Democratic Party") ;
questions[0][1][1].correct = 2 ;

strcpy(questions[0][1][2].q, "The ozone layer restricts") ;
strcpy(questions[0][1][2].options[0], "Visible light") ;
strcpy(questions[0][1][2].options[1], "Infrared radiation") ;
strcpy(questions[0][1][2].options[2], "X-rays and gamma rays") ;
strcpy(questions[0][1][2].options[3], "Ultraviolet radiation") ;
questions[0][1][2].correct = 4 ;

strcpy(questions[0][2][0].q, "Find the correct spelling") ;
```

```
53 strcpy(questions[0][2][0].options[0], "Treachrous") ;
54 strcpy(questions[0][2][0].options[1], "Trecherous") ;
55 strcpy(questions[0][2][0].options[2], "Trechearous") ;
56 strcpy(questions[0][2][0].options[3], "Treacherous") ;
57 questions[0][2][0].correct = 4 ;
58
59 strcpy(questions[0][2][1].q, "Find the synonym of Frugal") ;
60 strcpy(questions[0][2][1].options[0], "invention") ;
61 strcpy(questions[0][2][1].options[1], "economical") ;
62 strcpy(questions[0][2][1].options[2], "to whisper") ;
63 strcpy(questions[0][2][1].options[3], "explore") ;
64 questions[0][2][1].correct = 2 ;
65
66 strcpy(questions[0][2][2].q, "Complete the sentence: Despite his best efforts to
       conceal his anger ________") ;
67 strcpy(questions[0][2][2].options[0], "people came to know that he was annoyed")
       ;
68 strcpy(questions[0][2][2].options[1], "he failed to give us an impression of his
       agony") ;
69 strcpy(questions[0][2][2].options[2], "he succeeded in camouflaging his emotions"
       ) ;
70 strcpy(questions[0][2][2].options[3], "he could succeed in doing it easily") ;
71 questions[0][2][2].correct = 1 ;
72
73 strcpy(questions[1][0][0].q, "The pointer which always stores the current active
       object address is _____") ;
74 strcpy(questions[1][0][0].options[0], "auto_ptr") ;
75 strcpy(questions[1][0][0].options[1], "this") ;
76 strcpy(questions[1][0][0].options[2], "p") ;
77 strcpy(questions[1][0][0].options[3], "None of the above") ;
78 questions[1][0][0].correct = 2 ;
79
80 strcpy(questions[1][0][1].q, "Which of the following is a valid real constant in
       exponent form in C++?") ;
81 strcpy(questions[1][0][1].options[0], "172.E5") ;
82 strcpy(questions[1][0][1].options[1], "1.52E07") ;
83 strcpy(questions[1][0][1].options[2], "13,2E05") ;
84 strcpy(questions[1][0][1].options[3], "17.17E2.3") ;
85 questions[1][0][1].correct = 2 ;
86
87 strcpy(questions[1][0][2].q, "In C++, the header file required to use
       setpricision() is ______") ;
88 strcpy(questions[1][0][2].options[0], "<iostream.h>") ;
89 strcpy(questions[1][0][2].options[1], "<math.h>") ;
90 strcpy(questions[1][0][2].options[2], "<iomanip.h>") ;
91 strcpy(questions[1][0][2].options[3], "<utility.h>") ;
92 questions[1][0][2].correct = 3 ;
93
94 strcpy(questions[1][1][0].q, "Which of the following are the members of SAARC (
       South Asian Association for Regional Cooperation)?") ;
95 strcpy(questions[1][1][0].options[0], "Bhutan, Bangladesh, India and Pakistan") ;
96 strcpy(questions[1][1][0].options[1], "Bhutan, Bangladesh, the Maldives, Nepal,
       India, Pakistan, Afghanistan and Sri Lanka") ;
97 strcpy(questions[1][1][0].options[2], "Afghanistan, Pakistan, Thailand, Indonesia
       , Nepal and Sri Lanka") ;
98 strcpy(questions[1][1][0].options[3], "None of the above") ;
99 questions[1][1][0].correct = 2 ;
100
101 strcpy(questions[1][1][1].q, "India's first satellite is named after ______") ;
```

```
102 strcpy(questions[1][1][1].options[0], "Aryabhatta") ;
103 strcpy(questions[1][1][1].options[1], "Bhaskara II") ;
104 strcpy(questions[1][1][1].options[2], "Bhaskara I") ;
105 strcpy(questions[1][1][1].options[3], "Albert Einstein") ;
106 questions[1][1][1].correct = 1 ;
107
108 strcpy(questions[1][1][2].q, "In which year, terrorists crash two planes into New
        York's World Trade Centre on September 11 in a sequence of destruction?") ;
109 strcpy(questions[1][1][2].options[0], "2000") ;
110 strcpy(questions[1][1][2].options[1], "2001") ;
111 strcpy(questions[1][1][2].options[2], "2002") ;
112 strcpy(questions[1][1][2].options[3], "2003") ;
113 questions[1][1][2].correct = 2 ;
114
115 strcpy(questions[1][2][0].q, "What is the antonym of HUAGHTY?") ;
116 strcpy(questions[1][2][0].options[0], "Cowardly") ;
117 strcpy(questions[1][2][0].options[1], "Scared") ;
118 strcpy(questions[1][2][0].options[2], "Pitiable") ;
119 strcpy(questions[1][2][0].options[3], "Humble") ;
120 questions[1][2][0].correct = 4 ;
121
122 strcpy(questions[1][2][1].q, "Complete the statement: He is so lazy that he
        ------") ;
123 strcpy(questions[1][2][1].options[0], "cannot depend on others for getting his
        work done") ;
124 strcpy(questions[1][2][1].options[1], "dislike to postpone the work that he
        undertakes to do") ;
125 strcpy(questions[1][2][1].options[2], "cannot delay the schedule of completing
        the work") ;
126 strcpy(questions[1][2][1].options[3], "can seldom complete his work on time") ;
127 questions[1][2][1].correct = 4 ;
128
129 strcpy(questions[1][2][2].q, "Choose the correct spelling") ;
130 strcpy(questions[1][2][2].options[0], "Palate") ;
131 strcpy(questions[1][2][2].options[1], "Palet") ;
132 strcpy(questions[1][2][2].options[2], "Palete") ;
133 strcpy(questions[1][2][2].options[3], "Pelate") ;
134 questions[1][2][2].correct = 1 ;
135
136 strcpy(questions[2][0][0].q, "In C++, which of the statements is valid?") ;
137 strcpy(questions[2][0][0].options[0], "int &refs[10] ;") ;
138 strcpy(questions[2][0][0].options[1], "int a2[] = a ;") ;
139 strcpy(questions[2][0][0].options[2], "a + b = c ;") ;
140 strcpy(questions[2][0][0].options[3], "None of the above") ;
141 questions[2][0][0].correct = 4 ;
142
143 strcpy(questions[2][0][1].q, "Which of the following statement is correct?") ;
144 strcpy(questions[2][0][1].options[0], "A reference has to be de-referenced to
        access a value.") ;
145 strcpy(questions[2][0][1].options[1], "A reference does not need to be de-
        referenced to access a value.") ;
146 strcpy(questions[2][0][1].options[2], "A reference has to be double de-referenced
        to access a value.") ;
147 strcpy(questions[2][0][1].options[3], "Whether a reference should be de-
        referenced or not depends on the type of the reference.") ;
148 questions[2][0][1].correct = 2 ;
149
150 strcpy(questions[2][0][2].q, "Which of the following is correct?") ;
151 strcpy(questions[2][0][2].options[0], "A reference is declared using * operator."
```

```
    ) ;
strcpy(questions[2][0][2].options[1], "Once a reference variable has been defined
     to refer to a particular variable it can refer to any other variable.") ;
strcpy(questions[2][0][2].options[2], "A reference must always be initialized
    within classes.") ;
strcpy(questions[2][0][2].options[3], "A variable can have multiple references.")
     ;
questions[2][0][2].correct = 4 ;

strcpy(questions[2][1][0].q, "In which of the following years, the membership of
    the Security Council was increased from 11 to 15 (under Article 23)?") ;
strcpy(questions[2][1][0].options[0], "1960") ;
strcpy(questions[2][1][0].options[1], "1965") ;
strcpy(questions[2][1][0].options[2], "1972") ;
strcpy(questions[2][1][0].options[3], "1975") ;
questions[2][1][0].correct = 2 ;

strcpy(questions[2][1][1].q, "ICAO stands for ______") ;
strcpy(questions[2][1][1].options[0], "International Civil Aviation Organization"
    ) ;
strcpy(questions[2][1][1].options[1], "Indian Corporation of Agriculture
    Organization") ;
strcpy(questions[2][1][1].options[2], "Institute of Company of Accounts
    Organization") ;
strcpy(questions[2][1][1].options[3], "None of the above") ;
questions[2][1][1].correct = 1 ;

strcpy(questions[2][1][2].q, "India's first Technicolor film ____ in the early
    1950s was produced by ____") ;
strcpy(questions[2][1][2].options[0], "\'Jhansi Ki Rani\', Sohrab Modi") ;
strcpy(questions[2][1][2].options[1], "\'Jhansi Ki Rani\', Sir Syed Ahmed") ;
strcpy(questions[2][1][2].options[2], "\'Mirza Ghalib\', Sohrab Modi") ;
strcpy(questions[2][1][2].options[3], "\'Mirza Ghalib\', Sir Syed Ahmed") ;
questions[2][1][2].correct = 1 ;

strcpy(questions[2][2][0].q, "Choose the correct spelling: ") ;
strcpy(questions[2][2][0].options[0], "Vetarinary") ;
strcpy(questions[2][2][0].options[1], "Vetinary") ;
strcpy(questions[2][2][0].options[2], "Veteninary") ;
strcpy(questions[2][2][0].options[3], "Veterinary") ;
questions[2][2][0].correct = 4 ;

strcpy(questions[2][2][1].q, "Pick the synonym of FECUND: ") ;
strcpy(questions[2][2][1].options[0], "fertile") ;
strcpy(questions[2][2][1].options[1], "hostile") ;
strcpy(questions[2][2][1].options[2], "immature") ;
strcpy(questions[2][2][1].options[3], "entangle") ;
questions[2][2][1].correct = 1 ;

strcpy(questions[2][2][2].q, "Pick the most appropriate word to complete the
    sentence: These slums are disgrace ________ the civic authorities.") ;
strcpy(questions[2][2][2].options[0], "towards") ;
strcpy(questions[2][2][2].options[1], "on") ;
strcpy(questions[2][2][2].options[2], "for") ;
strcpy(questions[2][2][2].options[3], "to") ;
questions[2][2][2].correct = 4 ;

}
```

Python Script

*T*his script takes in questions from the text files and arranges them in a structure defined in ques.cpp

```
# Converts Q's in .txt to struct in .cpp
#
# Script to convert questions in a .txt file to a .cpp file where they are stored in a
    struct
# The questions are read from files:
#   1. q_lvl1.txt
#   2. q_lvl2.txt
#   3. q_lvl3.txt
#
# They are wrapped in a function init_ques() that initialises these questions in a
   question type variable called questions
# \sa question, init_ques()

ques_cpp = open("ques.cpp", 'w');

ques_cpp.write('/* Generated from script.py */ \n\n')

ques_cpp.write('#include "quiz.h"\n')
ques_cpp.write('#include <string.h>\n\n')
ques_cpp.write('question questions[3][3][3];\n\n')

ques_cpp.write('void init_ques()\n{\n\n')

for n in xrange(0,3):
    if n == 0:
        ques_txt = open("q_lvl1.txt");
    elif n == 1:
        ques_txt = open("q_lvl2.txt");
    elif n == 2:
        ques_txt = open("q_lvl3.txt");

    for i in xrange(0,3):
        for j in xrange(0,3):
            question = ques_txt.readline()
            w = "strcpy(questions[%d][%d][%d].q, %s) ; \n" % (n, i, j, question[0:len(
                question) - 1])
            ques_cpp.write(w)
            for k in xrange(0, 4):
                option = ques_txt.readline()
                ques_cpp.write('strcpy(questions[%d][%d][%d].options[%d], %s) ; \n' %
                    (n, i, j, k, option[0:len(option) - 1]))

            answer = ques_txt.readline()
            ques_cpp.write('questions[%d][%d][%d].correct = %s ; \n\n' % (n, i, j,
                answer[0:1]))
            blank = ques_txt.readline()

    ques_txt.close()

ques_cpp.write('}')
ques_cpp.close()
```

Plain text files

These files contain the questions used in the project

1. q_lvl1.txt

```
"Select the INCORRECT keyword in C++"
"asm"
"virtual"
"statics"
"float"
3

"String literal \"abc\" will be represented in the memory as _____"
"abc\\0"
"abc/0"
"abc|0"
"abc_0"
1

"Which of the following header files contains the exit () function to terminate
    the current program in C++?"
"<string.h>"
"<process.h>"
"<iomanip.h>"
"<ctype.h>"
2

"Entomology is the science that studies ______"
"Behavior of human beings"
"Insects"
"The origin and history of technical and scientific terms"
"The formation of rocks"
2

"Hitler party which came into power in 1933 is known as"
"Labour Party"
"Nazi Party"
"Ku—Klux—Klan"
"Democratic Party"
2

"The ozone layer restricts"
"Visible light"
"Infrared radiation"
"X—rays and gamma rays"
"Ultraviolet radiation"
4

"Find the correct spelling"
"Treachrous"
"Trecherous"
"Trechearous"
"Treacherous"
4

"Find the synonym of Frugal"
"invention"
```

```
"economical"
"to whisper"
"explore"
2

"Complete the sentence: Despite his best efforts to conceal his anger --------"
"people came to know that he was annoyed"
"he failed to give us an impression of his agony"
"he succeeded in camouflaging his emotions"
"he could succeed in doing it easily"
1
```

2. q_lvl2.txt

```
"The pointer which always stores the current active object address is _____"
"auto_ptr"
"this"
"p"
"None of the above"
2

"Which of the following is a valid real constant in exponent form in C++?"
"172.E5"
"1.52E07"
"13,2E05"
"17.17E2.3"
2

"In C++, the header file required to use setpricision() is ______"
"<iostream.h>"
"<math.h>"
"<iomanip.h>"
"<utility.h>"
3

"Which of the following are the members of SAARC (South Asian Association for
    Regional Cooperation)?"
"Bhutan, Bangladesh, India and Pakistan"
"Bhutan, Bangladesh, the Maldives, Nepal, India, Pakistan, Afghanistan and Sri
    Lanka"
"Afghanistan, Pakistan, Thailand, Indonesia, Nepal and Sri Lanka"
"None of the above"
2

"India's first satellite is named after ______"
"Aryabhatta"
"Bhaskara II"
"Bhaskara I"
"Albert Einstein"
1

"In which year, terrorists crash two planes into New York's World Trade Centre on
     September 11 in a sequence of destruction?"
"2000"
"2001"
"2002"
"2003"
2

"What is the antonym of HAUGHTY?"
"Cowardly"
"Scared"
"Pitiable"
"Humble"
4

"Complete the statement: He is so lazy that he ______"
"cannot depend on others for getting his work done"
"dislike to postpone the work that he undertakes to do"
"cannot delay the schedule of completing the work"
"can seldom complete his work on time"
```

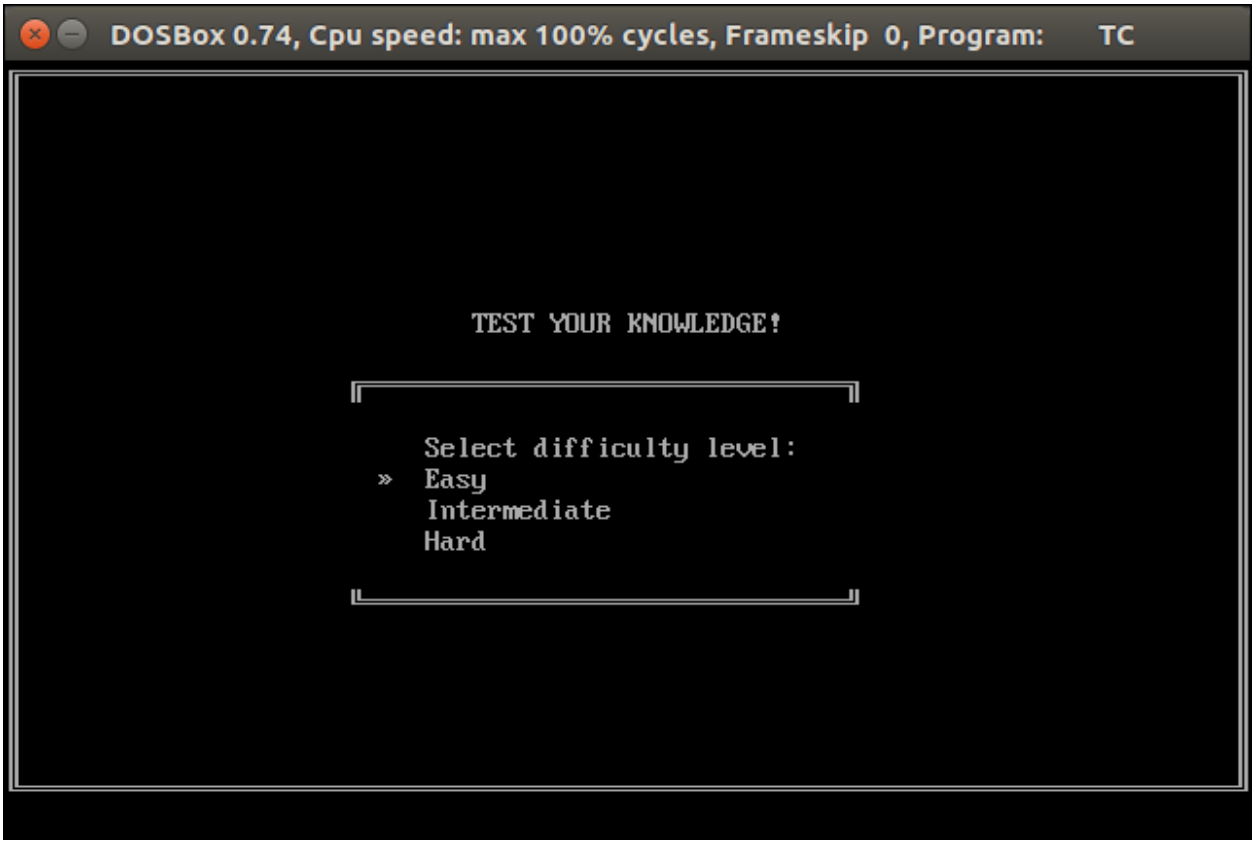
```
4

"Choose the correct spelling"
"Palate"
"Palet"
"Palete"
"Pelate"
1
```

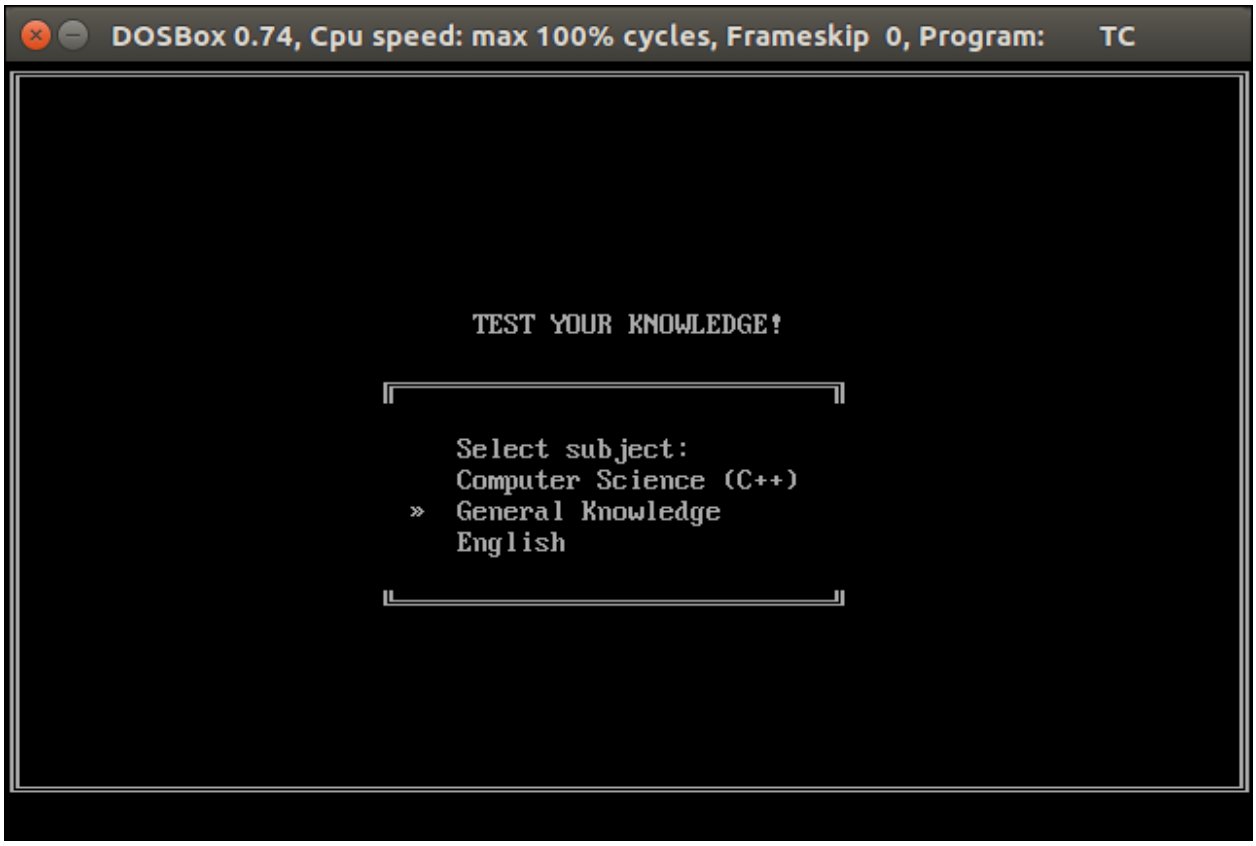
3. q_lvl3.txt

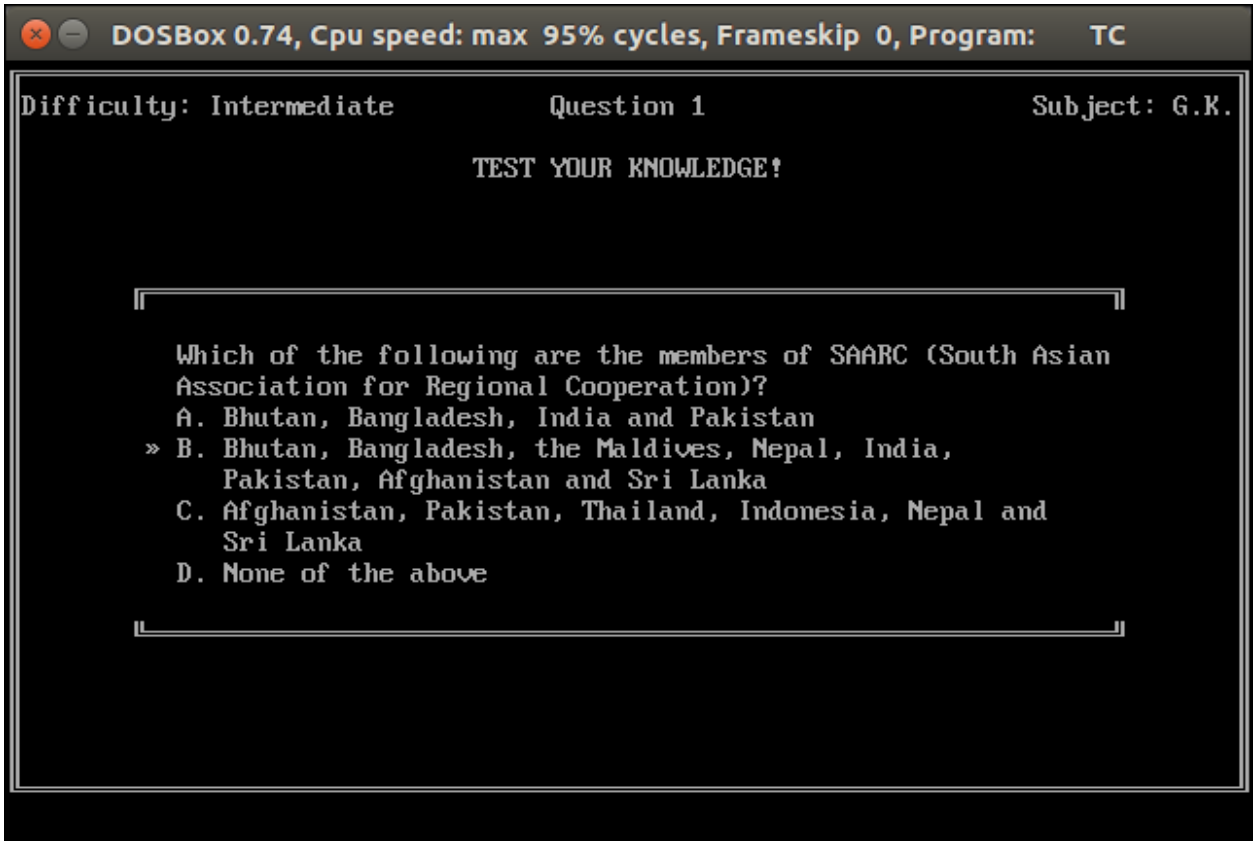
```
"In C++, which of the statements is valid?"
"int &refs[10] ;"
"int a2[] = a ;"
"a + b = c ;"
"None of the above"
4

"Which of the following statement is correct?"
"A reference has to be de—referenced to access a value."
"A reference does not need to be de—referenced to access a value."
"A reference has to be double de—referenced to access a value."
"Whether a reference should be de—referenced or not depends on the type of the reference."
2

"Which of the following is correct?"
"A reference is declared using * operator."
"Once a reference variable has been defined to refer to a particular variable it can refer to any other variable."
"A reference must always be initialized within classes."
"A variable can have multiple references."
4

"In which of the following years, the membership of the Security Council was increased from 11 to 15 (under Article 23)?"
"1960"
"1965"
"1972"
"1975"
2

"ICAO stands for ______"
"International Civil Aviation Organization"
"Indian Corporation of Agriculture Organization"
"Institute of Company of Accounts Organization"
"None of the above"
1

"India's first Technicolor film ____ in the early 1950s was produced by ____"
"\'Jhansi Ki Rani\', Sohrab Modi"
"\'Jhansi Ki Rani\', Sir Syed Ahmed"
"\'Mirza Ghalib\', Sohrab Modi"
"\'Mirza Ghalib\', Sir Syed Ahmed"
1

"Choose the correct spelling: "
"Vetarinary"
"Vetinary"
"Veteninary"
"Veterinary"
4

"Pick the synonym of FECUND: "
"fertile"
"hostile"
"immature"
"entangle"
```

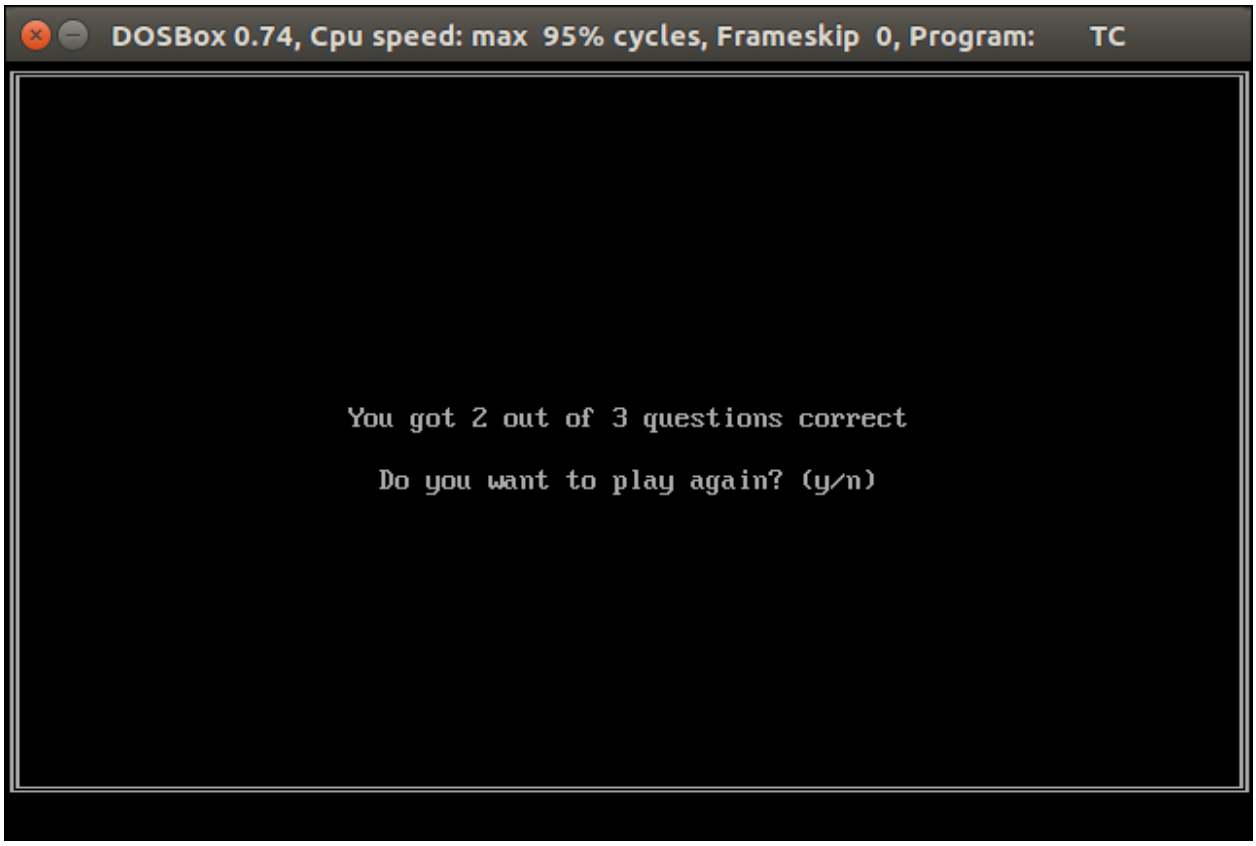
```
1

"Pick the most appropriate word to complete the sentence: These slums are
    disgrace -------- the civic authorities."
"towards"
"on"
"for"
"to"
4
```

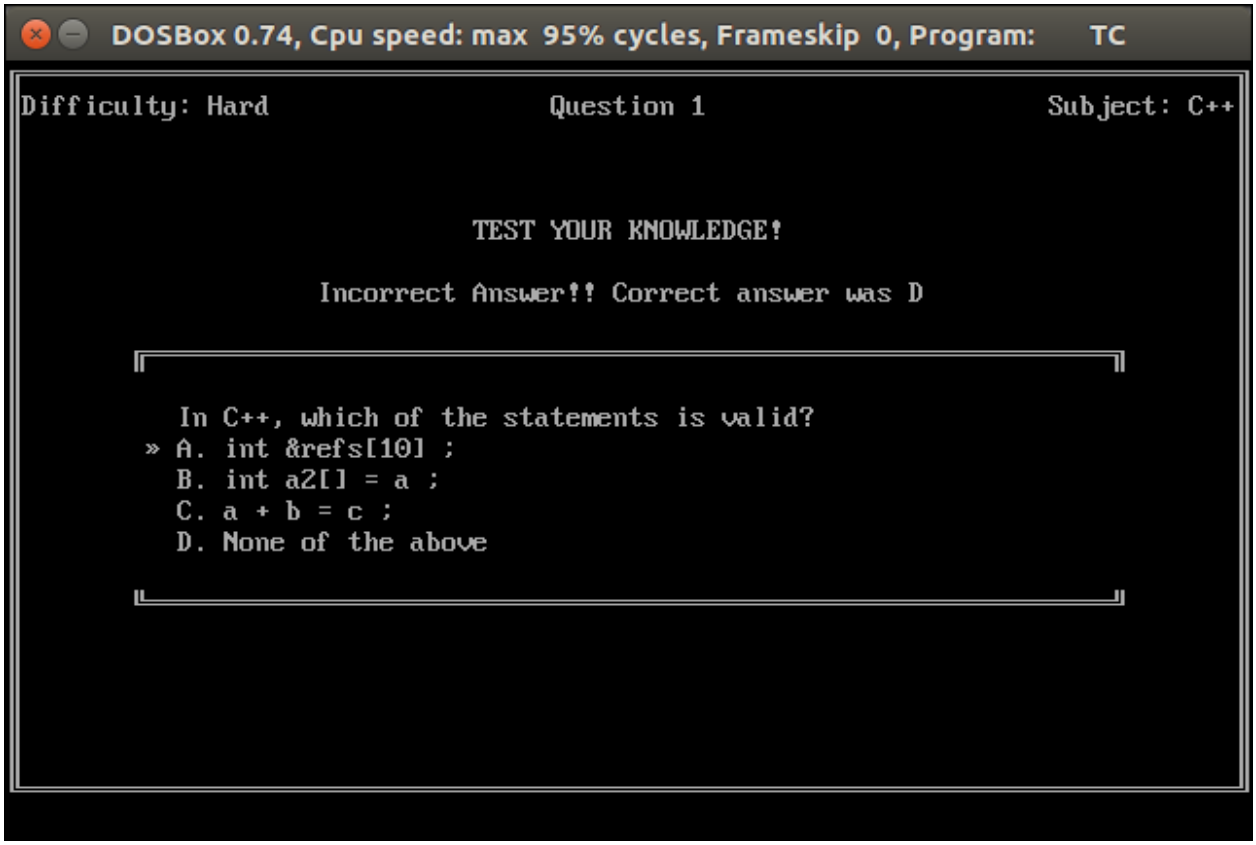
Output


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
TEST YOUR KNOWLEDGE!
Select difficulty level:
» Easy
Intermediate
Hard


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
TEST YOUR KNOWLEDGE!
Select subject:
Computer Science (C++)
» General Knowledge
English


DOSBox 0.74, Cpu speed: max 95% cycles, Frameskip 0, Program: TC
Difficulty: Intermediate Question 1 Subject: G.K.
TEST YOUR KNOWLEDGE!
Which of the following are the members of SAARC (South Asian Association for Regional Cooperation)?
A. Bhutan, Bangladesh, India and Pakistan
» B. Bhutan, Bangladesh, the Maldives, Nepal, India, Pakistan, Afghanistan and Sri Lanka
C. Afghanistan, Pakistan, Thailand, Indonesia, Nepal and Sri Lanka
D. None of the above


DOSBox 0.74, Cpu speed: max 95% cycles, Frameskip 0, Program: TC
You got 2 out of 3 questions correct
Do you want to play again? (y/n)


DOSBox 0.74, Cpu speed: max 95% cycles, Frameskip 0, Program: TC
Difficulty: Hard
Question 1
Subject: C++
TEST YOUR KNOWLEDGE!
Incorrect Answer!! Correct answer was D
In C++, which of the statements is valid?
» A. int &refs[10] ;
B. int a2[] = a ;
C. a + b = c ;
D. None of the above