```
/*C Program to implement Brute Force String Matching
Input  : 1. Text
         2. Pattern you want to search in the text
Output : 1. Location of the pattern in text – if search is successful
         2. If search is not successful -1
*/

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

int compareStrings(char [], char []);

int main() {
  char a[100], b[100];
  int position;

  printf("\n\n Enter some text: ");
  gets(a);

  printf("\n\n Enter a string you want to find in this text: ");
  gets(b);

  position = compareStrings(a, b);

  if(position != -1) {
    printf("\n\n Pattern is found at location %d\n\n", position + 1);
  }
  else {
    printf("\n\n Pattern does not exist in the text!\n\n");
  }
```

```
  return 0;
}

int compareStrings(char text[], char pattern[]) {
  int c, d, e, text_length, pattern_length, position = -1;

  text_length    = strlen(text);
  pattern_length = strlen(pattern);

  if (pattern_length > text_length) {
    return -1;
  }

  for (c = 0; c <= text_length - pattern_length; c++) {
    position = e = c;

    for (d = 0; d < pattern_length; d++) {
      if (pattern[d] == text[e]) {
        e++;
      }
      else {
        break;
      }
    }
    if (d == pattern_length) {
      return position;
    }
  }

  return -1;
}
```

Sample Input and Output:

1.

```
Enter some text: PERFECT_PROOF_LIES_ONLY_IN_MATHS

Enter a string you want to find in this text: PROOF

Pattern is found at location 9
Press any key to continue...
```

2.

```
Enter some text: PROGRAMMING_IS_EXCITING

Enter a string you want to find in this text: GRAMOPHONE

Pattern does not exist in the text!
Press any key to continue...
```