

## Project Candy Crush

Submission Dead Line: 30/12/2021







**Submission Instructions:** Make a folder, which name will be your Roll-Numbers in format (19L-1234 19L-1234 19L-1234 19L-1234), and place all (.cpp) code files and (.txt) files in that folder, then zip the folder and submit.










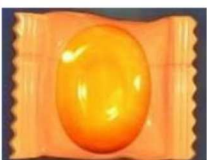
You will implement a single player game “Candy Crush” in C++. Here, a 2-D grid will be used for candies storage and the minimum size of grid will be 9×9.

At start of the game the whole grid will be filled with the new candies by randomly selecting them. The user must be provided with swapping options, so that he can swap two candies, if they lie side by side in same column, or row of grid. The target goal points and number of moves (swaps) should be set at start.

Type of Candies and their corresponding points are listed below.



Plain Candies	Points gained when crushed
 <b>Red</b> - Jelly Bean	30 Points
 <b>Yellow</b> - Lemon Drop	30 Points
 <b>Green</b> - Gum Square	40 Points
 <b>Blue</b> - Lollipop Head	50 Points
 <b>Orange</b> - Lozenge	60 Points
<b>Special Candies</b>	<b>Color Bomb</b>
	

Striped Candies				
				
Wrapped Candies				
				

## Game Rules:

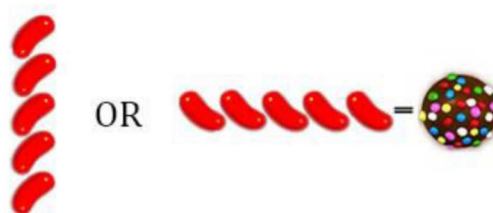
- Crush Plain Candies:** When three similar candies lie in same column, or row of grid, they will be getting crushed and empty slots of grid will be refilled with new candies by shifting candies from top towards bottom.
- Striped Candy:** It is created when 4 similar plain candies lie in same column, or row of grid. Striped candy can be made with 5 different colors Blue, Green, Orange, Red and Yellow.



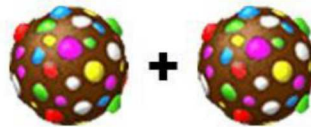
- Wrapped Candy:** It is created by matching 5 candies of same color in L or T shape. Wrapped candy can be made with 5 different colors Blue, Green, Orange, Red and Yellow.



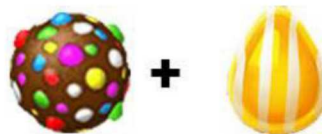
- Color Bomb:** It is created by matching 5 candies of same color in row or column.



5. **Plain Candy + Striped Candy:** When 3 similar color candies lie in same column, or row of grid, they will be crushed. If any of them is a striped candy then the corresponding rows or columns, which contains the striped candies, will be cleared. The decision to clear the row or column will be made according to direction of stripes on candy.
6. **Plain Candy + Wrapped candy:** When 3 similar color candies lie in same column, or row of grid, they will be crushed. If any of them is a wrapped candy then the candies in (3 x 3) square area of board around wrapped candy will also be crushed.
7. **Color Bomb + Plain Candy:** If player swaps color bomb with any plain candy, then all candies, of that particular color, will be crushed.
8. **Color Bomb + Color Bomb:** If player swaps color bomb with another color bomb, then all candies from board will be crushed and board will be reloaded with all new candies.



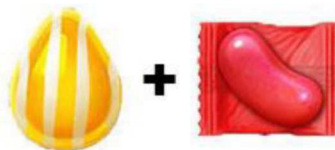
9. **Striped Candy + Color Bomb:** If player swaps color bomb with striped candy, then all candies will be transformed into striped candies of same color as of striped candy. All of them then clear their respective rows/columns.



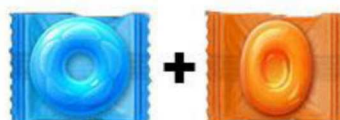
10. **Wrapped Candy + Color Bomb:** If player swaps color bomb with wrapped candy, then all candies will be transformed into wrapped candies of same color of wrapped candy. All candies in (3 x 3) square area, around swapped wrapped candy, will also be crushed.



11. **Striped Candy + Wrapped Candy:** If player swaps striped candy with wrapped candy, then 3 rows and 3 columns centered on the candy that was swapped will be cleared.



12. **Wrapped Candy + Wrapped Candy:** If player swaps wrapped candy with another wrapped candy, then all candies will be crushed in (5 x 5) square area around swapped candies.



13. **Winner/Loser:** The winner of this game is decided on the basis of points gained in 20 moves. If gained pointes are below targeted goal points then player will lose the game.

- 14. User Interaction:** Design an interactive graphical user interface for game. Whenever a new game is started, number and placement of candies should be random. Your game should be user interactive and should provide proper options to player. The information of target point, gained points and remaining moves should be updated accordingly.
- 15. Save\Load Game:** A player can stop game in middle and can continue next time with that game. Therefore, you have to store the state of Game in a file, and next time user will be provided with two options, load new game or play the previous one.

HAPPY PROGRAMMING