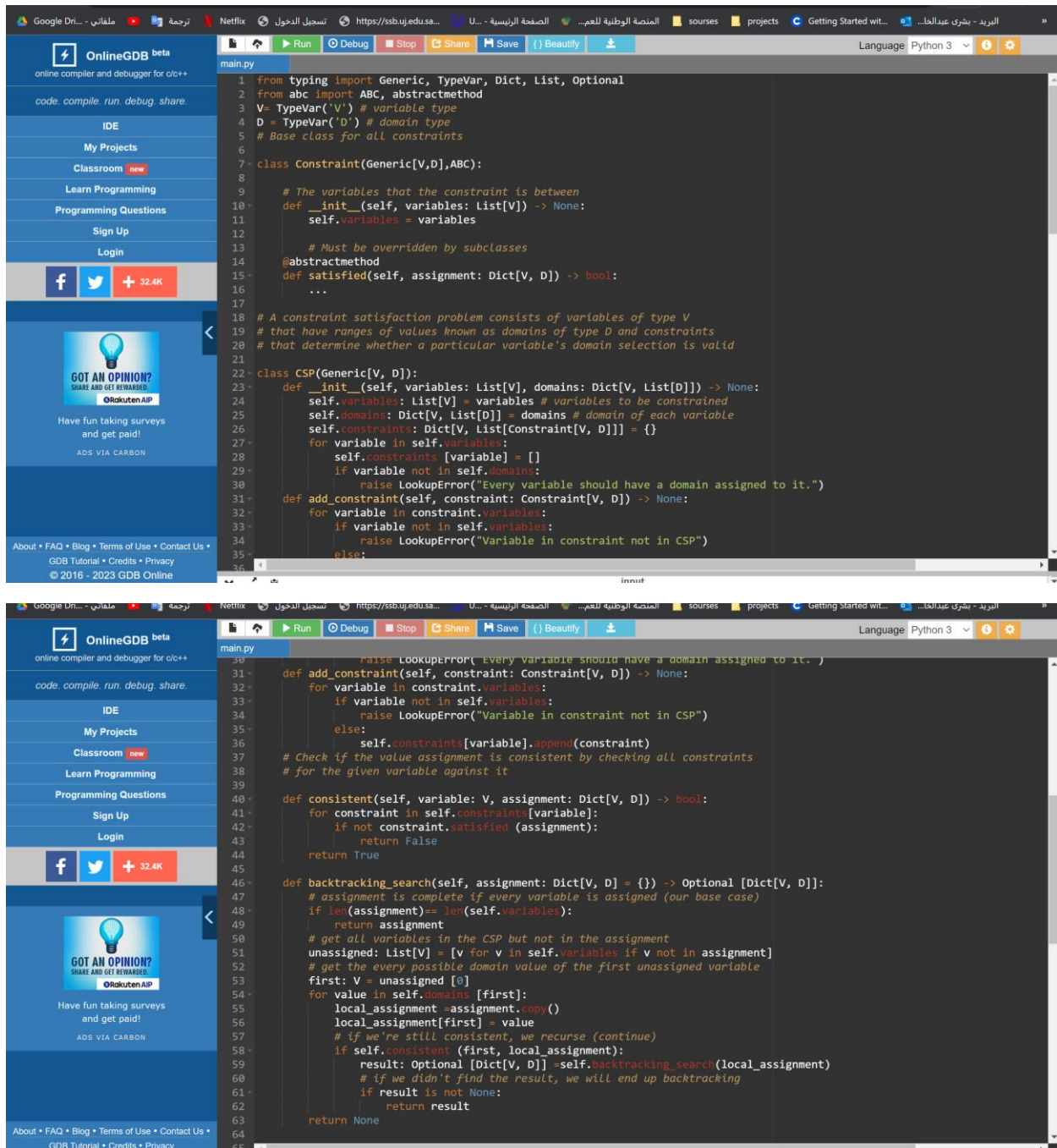


Lab8

Bushra dajam

2110054

Code:



The screenshot displays the OnlineGDB beta web interface. The left sidebar contains navigation links: 'code.compile.run.debug.share.', 'IDE', 'My Projects', 'Classroom' (marked as new), 'Learn Programming', 'Programming Questions', 'Sign Up', and 'Login'. Below these are social media icons for Facebook, Twitter, and a '+ 32.4K' button. A 'GOT AN OPINION?' survey by OrkutenaAP is also visible. The main area shows a Python 3 code editor with the following code:

```
1 from typing import Generic, TypeVar, Dict, List, Optional
2 from abc import ABC, abstractmethod
3 V = TypeVar('V') # variable type
4 D = TypeVar('D') # domain type
5 # Base class for all constraints
6
7 class Constraint(Generic[V,D],ABC):
8
9     # The variables that the constraint is between
10     def __init__(self, variables: List[V]) -> None:
11         self.variables = variables
12
13     # Must be overridden by subclasses
14     @abstractmethod
15     def satisfied(self, assignment: Dict[V, D]) -> bool:
16         ...
17
18 # A constraint satisfaction problem consists of variables of type V
19 # that have ranges of values known as domains of type D and constraints
20 # that determine whether a particular variable's domain selection is valid
21
22 class CSP(Generic[V, D]):
23     def __init__(self, variables: List[V], domains: Dict[V, List[D]]) -> None:
24         self.variables: List[V] = variables # variables to be constrained
25         self.domains: Dict[V, List[D]] = domains # domain of each variable
26         self.constraints: Dict[V, List[Constraint[V, D]]] = {}
27         for variable in self.variables:
28             self.constraints [variable] = []
29             if variable not in self.domains:
30                 raise LookupError("Every variable should have a domain assigned to it.")
31     def add_constraint(self, constraint: Constraint[V, D]) -> None:
32         for variable in constraint.variables:
33             if variable not in self.variables:
34                 raise LookupError("Variable in constraint not in CSP")
35             else:
36                 self.constraints [variable].append(constraint)
37
38 # Check if the value assignment is consistent by checking all constraints
39 # for the given variable against it
40
41 def consistent(self, variable: V, assignment: Dict[V, D]) -> bool:
42     for constraint in self.constraints [variable]:
43         if not constraint.satisfied (assignment):
44             return False
45     return True
46
47 def backtracking_search(self, assignment: Dict[V, D] = {}) -> Optional [Dict[V, D]]:
48     # assignment is complete if every variable is assigned (our base case)
49     if len(assignment) == len(self.variables):
50         return assignment
51     # get all variables in the CSP but not in the assignment
52     unassigned: List[V] = [v for v in self.variables if v not in assignment]
53     # get the every possible domain value of the first unassigned variable
54     first: V = unassigned [0]
55     for value in self.domains [first]:
56         local_assignment = assignment.copy()
57         local_assignment [first] = value
58         # if we're still consistent, we recurse (continue)
59         if self.consistent (first, local_assignment):
60             result: Optional [Dict[V, D]] =self.backtracking_search(local_assignment)
61             # if we didn't find the result, we will end up backtracking
62             if result is not None:
63                 return result
64     return None
```

```
OnlineGDB beta
online compiler and debugger for c/c++
code compile run debug share.


IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
f + 32.4K

GOT AN OPINION?
SHARE AND GET REWARDED.
OkutenAP
Have fun taking surveys
and get paid!
ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us •
GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online

main.py
59 result: Optional[Dict[V, D]] = self.backtracking_search(local_assignment)
60 # if we didn't find the result, we will end up backtracking
61 if result is not None:
62     return result
63 return None
64
65 from typing import Dict, List, Optional
66
67 class MapColoringConstraint(Constraint[str, str]):
68     def __init__(self, place1: str, place2: str) -> None:
69         super().__init__([place1, place2])
70         self.place1: str = place1
71         self.place2: str = place2
72     def satisfied(self, assignment: Dict[str, str]) -> bool:
73         # If either place is not in the assignment then it is not
74         # yet possible for their colors to be conflicting
75         if self.place1 not in assignment or self.place2 not in assignment:
76             return True
77         # check the color assigned to place1 is not the same as the
78         # color assigned to place2
79         return assignment[self.place1] != assignment[self.place2]
80
81 if __name__ == "__main__":
82     variables: List[str] = ["Western Australia", "Northern Territory", "South Australia", "Queensland", "New South Wales", "Victoria", "Tasmania"]
83     domains: Dict[str, List[str]] = {}
84     for variable in variables:
85         domains[variable] = ["red", "green", "blue"]
86     csp: CSP[str, str] = CSP(variables, domains)
87     csp.add_constraint(MapColoringConstraint("Western Australia", "South Australia"))
88     csp.add_constraint(MapColoringConstraint("Queensland", "South Australia"))
89     csp.add_constraint(MapColoringConstraint("Victoria", "South Australia"))
90     csp.add_constraint(MapColoringConstraint("Western Australia", "Northern Territory"))
91     csp.add_constraint(MapColoringConstraint("South Australia", "Northern Territory"))
92     csp.add_constraint(MapColoringConstraint("Queensland", "Northern Territory"))
93     csp.add_constraint(MapColoringConstraint("Queensland", "New South Wales"))
94
95     solution: Optional[Dict[str, str]] = csp.backtracking_search()
96
97     if solution is None:
98         print("No solution found!")
99     else:
100         print(solution)
```

Output:

 OnlineGDB beta

online compiler and debugger for c/c++

code, compile, run, debug, share.

IDE

My Projects




Classroom new


Learn Programming


Programming Questions

Sign Up

Login

   32.4K

 GOT AN OPINION?
SHARE AND GET REWARDED.



Have fun taking surveys
and get paid!

ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us •

main.py

Run Debug Stop Share Save Beauty

Language Python 3

```
79         return assignment[self.place1] != assignment[self.place2]
80
81 if __name__ == "__main__":
82     variables: List[str] = ["Western Australia", "Northern Territory", "South Australia", "Queensland", "New South Wales", "Victoria", "Tasmania"]
83     domains: Dict[str, List[str]] = {}
84     for variable in variables:
85         domains[variable] = ["red", "green", "blue"]
86     csp: CSP[str, str] = CSP(variables, domains)
87     csp.add_constraint(MapColoringConstraint("Western Australia", "South Australia"))
88     csp.add_constraint(MapColoringConstraint("Queensland", "South Australia"))
89     csp.add_constraint(MapColoringConstraint("Victoria", "South Australia"))
90     csp.add_constraint(MapColoringConstraint("Western Australia", "Northern Territory"))
91     csp.add_constraint(MapColoringConstraint("South Australia", "Northern Territory"))
92     csp.add_constraint(MapColoringConstraint("Queensland", "Northern Territory"))
93     csp.add_constraint(MapColoringConstraint("Queensland", "New South Wales"))
94     csp.add_constraint(MapColoringConstraint("New South Wales", "South Australia"))
95     csp.add_constraint(MapColoringConstraint("Victoria", "New South Wales"))
96
97 input
98 {'Western Australia': 'red', 'Northern Territory': 'green', 'South Australia': 'blue', 'Queensland': 'red', 'New South Wales': 'green', 'Victoria': 'red', 'Tasmania': 'green'}
99
100 ...Program finished with exit code 0
101 Press ENTER to exit console.
```