# Lab – 3 Selections

Tasks:

1. Rewrite the following program using chained conditionals (i.e. without using nested if):

age = int(input("Input age: "))

if age>=6:

if age>12:

print( "Adult Ticket" )

else:

print( "Child Ticket" )

else:

print( "Free of charge" )

Answer:

age = int(input("Input age: "))

if age>12:

print( "Adult Ticket" )

elif age >=6

print( "Child Ticket" )

else:

print( "Free of charge" )

1. Rewrite the following program without using logical operator (i.e. and, or, not)

degree = int(input("Input degree value: "))

if degree < 0:

print( "Freezing" )

elif degree > 30 and degree < 100:

print( "Very hot" )

elif degree >=0 and degree < 20:

print( "Cold" )

elif degree >=100:

print( "Boiling" )

elif degree >=20 and degree <= 30:

print( "OK" )

Answer:

degree = int(input("Input degree value: "))

if degree < 0:

print( "Freezing" )

elif degree < 20:

print( "Cold" )

elif degree <= 30:

print( "OK" )

elif degree < 100

print ( "Very hot" )

elif degree >= 100

print ( "Boiling" )

1. Assume that two Boolean variables raining and have\_umbrella are properly declared and initialized. Display the corresponding message depending on the values of the two variables stated in the table below. Do not use nested if.

|  |  |  |
| --- | --- | --- |
| Value of raining | Value of have\_umbrella | Corresponding Message |
| True | True | Bring your umbrella |
| True | False | Stay home |
| False | True | Put away your umbrella |
| False | False |

raining = input("Is raining? (Y/N): ")

if raining.strip().upper() == "Y":

raining = True

else:

raining = False

have\_umbrella = input("Have Umbrella? (Y/N): ")

if have\_umbrella.strip().upper() == "Y":

have\_umbrella = True

else:

have\_umbrella = False

# your code starts here

Output:

User’s input

Is raining? (Y/N): Y

Have Umbrella? (Y/N): N

User’s input

Stay home

Answer:

raining = input("Is raining? (Y/N): ")

if raining.strip().upper() == "Y":

raining = True

else:

raining = False

have\_umbrella = input("Have Umbrella? (Y/N): ")

if have\_umbrella.strip().upper() == "Y":

have\_umbrella = True

else:

have\_umbrella = False

# your code starts here

if raining == True and have\_umbrella == True:

print ("Bring your umbrella")

elif raining == True and have\_umbrella == False:

print ("Stay home")

elif raining == False:

print ("Put away your umbrella")

1. Write a program to display a student’s grade by inputting student’s mark according to the following table. Do not use any logical operator (i.e. and, or, not) in your code.

|  |  |
| --- | --- |
| Marks | Grade |
| 0 – 39 | F |
| 40 – 49 | D |
| 50 – 59 | C |
| 60 – 69 | B |
| 70 – 100 | A |

Output:

Input student’s mark: 55

User’s input

This student obtained grade C

Answer:

marks = int(input("Input student's mark: "))

if marks < 40:

print("This student obtained grade F")

elif marks < 50:

print("This student obtained grade D")

elif marks < 60:

print("This student obtained grade C")

elif marks < 70:

print("This student obtained grade B")

else:

print("This student obtained grade A")

1. Write a program to display the corresponding message according to the following table. Do not use any logical operator (i.e. and, or, not) in your code.

|  |  |
| --- | --- |
| Range of cost | Corresponding Message |
| > 100 | Expensive |
| 51 – 100 | OK to buy |
| 20 – 50 | Cheap |
| < 20 | Be careful |

Output:

User’s input

Input cost: 55

OK to buy

Answer:

cost = int(input("Input cost: "))

if cost > 100:

print("Expensive")

elif cost > 50:

print("OK to buy")

elif cost >= 20:

print("Cheap")

elif cost < 20:

print("Be careful")

1. Write a program to calculate the two roots of a quadratic equation () by the formulas:

The discriminant (Δ), can be used to determine the number of roots. Your program should prompt the user to enter the values of a, b, c and handle the following cases:

|  |  |  |
| --- | --- | --- |
| Cases | Description | Output |
| a = 0 | not a quadratic equation | This is not a quadratic equation |
| Δ > 0 | two roots | 2 roots, x1=…, x2=… |
| Δ = 0 | one roots | 1 root, x=… |
| Δ < 0 | no real root | No real roots |

Test your program with the following six test cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | Quadratic Equation | Input values of a, b, c | Discriminant | Expected Output |
| 1 | *x* – 1 = 0 | a=0, b=1, c=-1 | n/a | This is not a quadratic equation |
| 2 | *x*2 – 1 = 0 | a=1, b=0, c=-1 | 4 | 2 roots, x1=1, x2=-1 |
| 3 | *x*2 – *x* = 0 | a=1, b=-1, c=0 | 1 | 2 roots, x1=1, x2=0 |
| 4 | *x*2 – 2*x* + 1 = 0 | a=1, b=-2, c=1 | 0 | 1 root, x=1 |
| 5 | *x*2 + *x* + 1 = 0 | a=1, b=1, c=1 | -3 | No real roots |
| 6 | 3*x*2 – 4*x* + 1 = 0 | a=3, b=-4, c=1 | 4 | 2 roots, x1=1, x2=0.3333333333 |

Save the program as **lab03\_q6.py**

Answer:

a = float(input("Input the value a: "))

b = float(input("Input the value b: "))

c = float(input("Input the value c: "))

if a==0:

print("This is not a quadratic equation")

else:

discriminant = (b \* b - 4 \* a \* c)

if discriminant > 0: # two roots

x1= (-b + discriminant \*\* 0.5) / (2 \* a)

x2= (-b - discriminant \*\* 0.5) / (2 \* a)

print ("2 roots, x1 =",x1,", x2 =",x2)

elif discriminant == 0: # one root:

x= -b / (2 \* a)

print ("1 root, x =",x)

else: # no roots

print ("No real roots")

1. Write a program to ask user for the year and month, it then display the number of days in that month. Test your program with the following sets of data:

|  |  |  |
| --- | --- | --- |
| Input Values | | Output |
| Year | Month | #days |
| 1995 | 2 | 28 |
| 1996 | 2 | 29 |
| 2000 | 2 | 29 |
| 2017 | 8 | 31 |
| 2020 | 11 | 30 |
| 2100 | 2 | 28 |

Output:

Enter Year: 2000

User’s input

Enter Month: 2

Number of days in this month: 29

Answer:

year = int(input("Enter Year: "))

month = int(input("Enter Month: "))

if month == 2:

if year % 400 == 0 or year % 4 == 0 and year % 100 != 0:

days = 29

else:

days = 28

elif month % 2 == 1 and month <= 7 or month % 2 == 0 and month >= 8:

days = 31

else:

days = 30

print ("Number of days in this month:", days)