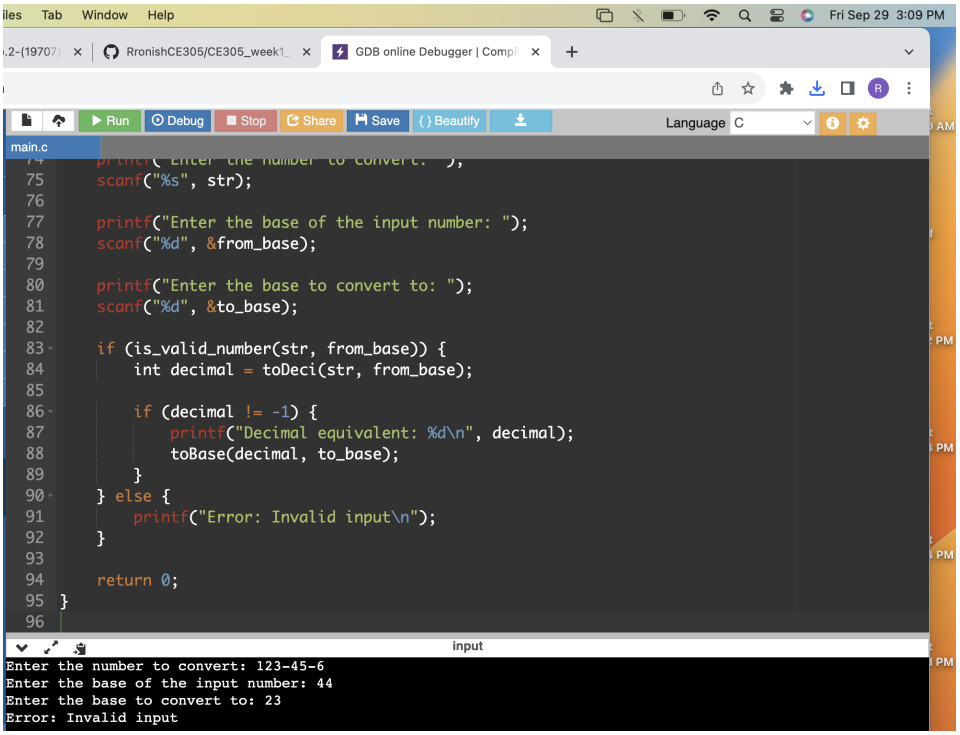
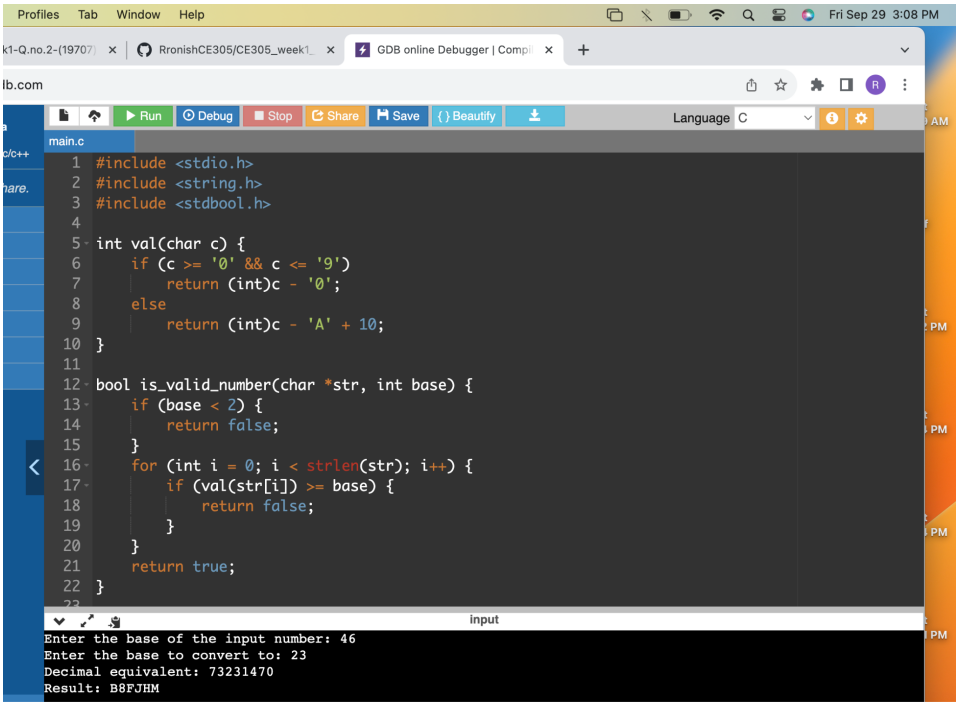


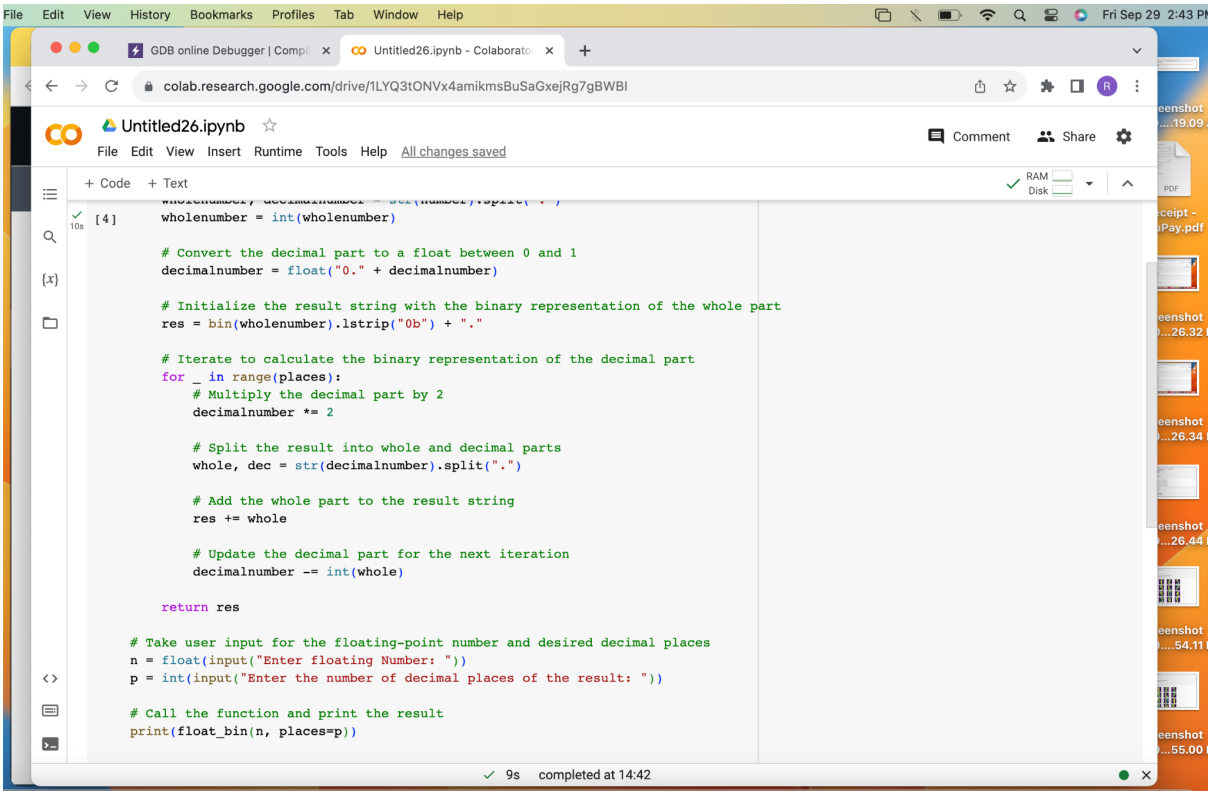
```c
    scanf("%s", str);

    printf("Enter the base of the input number: ");
    scanf("%d", &from_base);

    printf("Enter the base to convert to: ");
    scanf("%d", &to_base);

    if (is_valid_number(str, from_base)) {
        int decimal = toDeci(str, from_base);

        if (decimal != -1) {
            printf("Decimal equivalent: %d\n", decimal);
            toBase(decimal, to_base);
        }
    } else {
        printf("Error: Invalid input\n");
    }

    return 0;
}
```

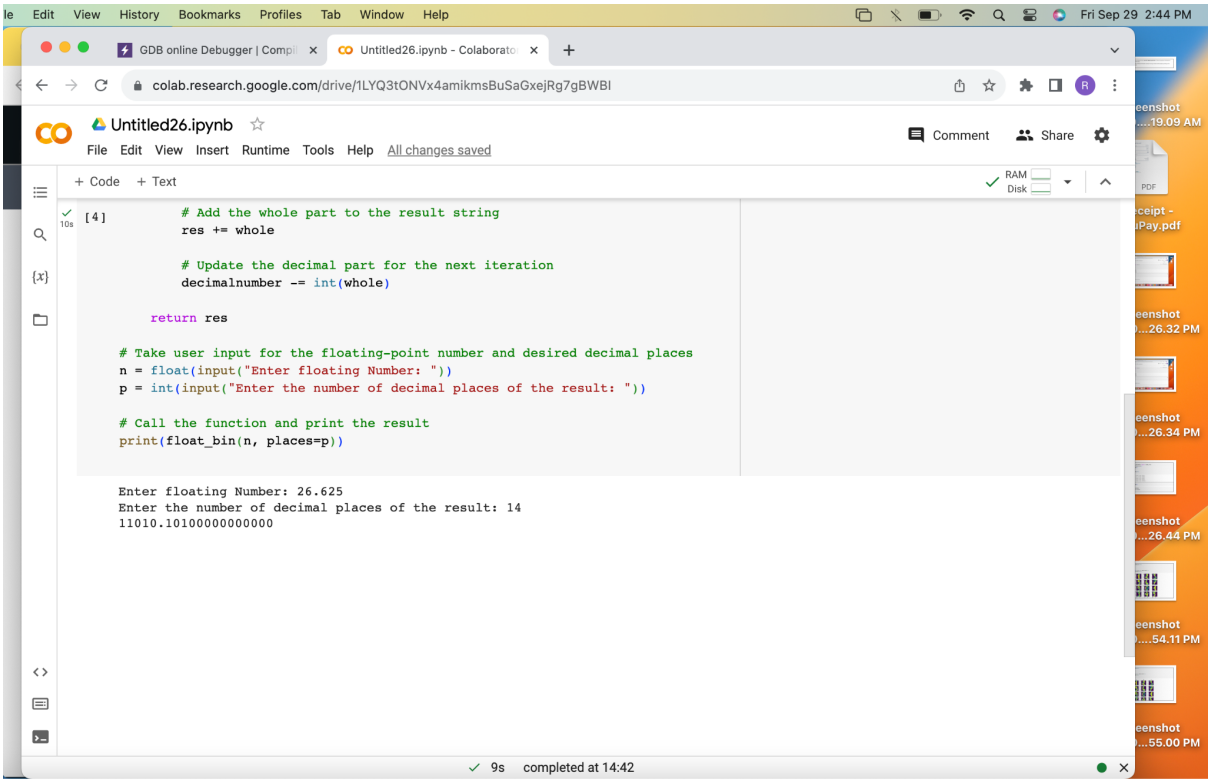
```
Enter the number to convert: 123-45-6
Enter the base of the input number: 44
Enter the base to convert to: 23
Error: Invalid input
```



```c
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

int val(char c) {
    if (c >= '0' && c <= '9')
        return (int)c - '0';
    else
        return (int)c - 'A' + 10;
}

bool is_valid_number(char *str, int base) {
    if (base < 2) {
        return false;
    }
    for (int i = 0; i < strlen(str); i++) {
        if (val(str[i]) >= base) {
            return false;
        }
    }
    return true;
}
```

```
Enter the base of the input number: 46
Enter the base to convert to: 23
Decimal equivalent: 73231470
Result: B8FJHM
```


File Edit View History Bookmarks Profiles Tab Window Help
Fri Sep 29 2:43 PM
GDB online Debugger | Compl
Untitled26.ipynb - Colaborato
colab.research.google.com/drive/1LYQ3tONVx4amikmsBuSaGxejRg7gBWBI
Untitled26.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[4]
wholenumber = int(wholenumber)
# Convert the decimal part to a float between 0 and 1
decimalnumber = float("0." + decimalnumber)
# Initialize the result string with the binary representation of the whole part
res = bin(wholenumber).lstrip("0b") + "."
# Iterate to calculate the binary representation of the decimal part
for _ in range(places):
# Multiply the decimal part by 2
decimalnumber *= 2
Split the result into whole and decimal parts
whole, dec = str(decimalnumber).split(".")
Add the whole part to the result string
res += whole
Update the decimal part for the next iteration
decimalnumber -= int(whole)
return res
# Take user input for the floating-point number and desired decimal places
n = float(input("Enter floating Number: "))
p = int(input("Enter the number of decimal places of the result: "))
# Call the function and print the result
print(float_bin(n, places=p))
9s completed at 14:42


Edit View History Bookmarks Profiles Tab Window Help
Fri Sep 29 2:44 PM
GDB online Debugger | Compl
Untitled26.ipynb - Colaborato
colab.research.google.com/drive/1LYQ3tONVx4amikmsBuSaGxejRg7gBWBI
Untitled26.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[4]
Add the whole part to the result string
res += whole
Update the decimal part for the next iteration
decimalnumber -= int(whole)
return res
# Take user input for the floating-point number and desired decimal places
n = float(input("Enter floating Number: "))
p = int(input("Enter the number of decimal places of the result: "))
# Call the function and print the result
print(float_bin(n, places=p))
Enter floating Number: 26.625
Enter the number of decimal places of the result: 14
11010.10100000000000
9s completed at 14:42