```
############################## Running a question:

def run_question(q):
    for x, y in q.items():
        if x in ["question"]:
            print("{}".format(y))
        if x in ["A.", "B."]:
            print("{}{}".format(x, y))
    answer = get_single_input()
    if answer == "A.":
        bernie_count += 1
    elif answer == "B.":
        joe_count += 1

############################## Getting input:

def get_single_input():
    get_input = True
    while get_input == True:
        user_input = input("Please enter a letter: ")
        user_input = user_input.upper()
        user_input = user_input.strip()
        if len(user_input)!=1:
            print("You may only enter one character.")
            continue
        if user_input not in ["A", "B"]:
            print("Sorry, that character was invalid!")
            continue
        get_input = False
    return user_input

run_question(question_one)

if bernie_count > joe_count:
    result = Bernie
    print("Your result was Bernie Sanders!")
elif joe_count > bernie_count:
    result = Joe
    print("Your result was Joe Biden.")
```

you are testing for "A." but the get_single_input will return only "A" or "B"

So these conditions will never execute

I know I did this but

```
if ...
elif ...
else:
    get_input = False
```

is nicer

what's going to happen here ???

else:

it is always good to have an "else" at the end

```
else:
    print("This is unexpected")
```

or "It's a tie" or something.

This guarantees that you always get a response.