

```

def print_header():
    # show the title and intro text
    print("=" * 60) # top border
    print(" " * 15 + "BEYOND THE BLUES") # centered title
    print(" " * 12 + "Am I Depressed?") # centered subtitle
    print("=" * 60) # bottom border
    print("\nThis tool helps you check your mental health.") # explain what this is
    print("Answer honestly. No one will see this but you.\n") # reassure them it's private
    print("=" * 60 + "\n") # separator

def get_user_info():
    # ask for basic info about the person
    print("PART 1: About You")
    print("-" * 40) # section divider

    name = input("Name: ") # get their name
    age = int(input("Age: ")) # get their age and convert to number

    # let them pick their gender from options
    gen_opts = {'1': 'Male', '2': 'Female', '3': 'Non-binary/Other', '4': 'Prefer not to say'}

    print("\nGender:")
    for k in gen_opts:
        print(k + ". " + gen_opts[k])

    gencho = input("Pick one (1-4): ") # get their choice
    gen = gen_opts.get(gencho, 'Prefer not to say') # match choice to gender, default if invalid

    # figure out their category based on age
    if age < 18:
        age_grp = "Below 18"
    else:
        age_grp = "18 or older"
    cat = gen + " (" + age_grp + ")" # combine gender and age group

    # ask how long they've been feeling this way
    dur_opts = {'1': 'Less than 2 weeks', '2': '2 weeks to 1 month', '3': '1 to 3 months', '4': 'More than 3 months'}

    print("\nHow long have you felt this way?")
    for k in dur_opts:
        print(k + ". " + dur_opts[k])

    durcho = input("Pick one (1-4): ") # get their choice

```

```

dur = dur_opts.get(durcho, 'More than 3 months') # match choice to duration

# check if they talked to a professional before
print("\nEver talked to a professional about mental health?")
print("1. Yes\n2. No")
prev_input = input("Pick one (1-2): ")
if prev_input == '1':
    prev = "Yes"
else:
    prev = "No"

print("\n" + "=" * 60 + "\n") # end of section
return name, age, gen, cat, dur, prev # send back all the info we collected

def ask_q(q, num):
    # ask a single question and get the answer
    print("\nQ" + str(num) + ": " + q) # show question number and text
    opts = ["0. Never", "1. Rarely", "2. Sometimes", "3. Often", "4. Always"] # answer choices
    for o in opts: # show each option
        print(o)
    ans = input("Answer (0-4): ") # get their answer
    return int(ans) # convert to number and return

def get_qs(gen, age):
    # store all the questions organized by type, gender, and age
    qs = {
        'behavioral': {
            'Male': {
                'under18': [
                    "How often do you skip school or skip classes?",
                    "Do you have trouble focusing on homework or gaming?",
                    "How often do you avoid your friends or group chats?",
                    "Do you sleep through your alarm or stay in bed all day?",
                    "How often do you feel lazy to do anything, even stuff you like?"
                ],
                'adult': [
                    "How often do you feel overwhelmed by work or daily tasks?",
                    "Do you struggle to focus or be productive at work?",
                    "How often do you cancel plans or avoid social gatherings?",
                    "Do you find it hard to get up and start your day?",
                    "How often do you experience changes in your sleep schedule?"
                ]
            },
            'Female': {

```

```
'under18': [
    "How often do you avoid going to school or joining activities?",
    "Do you have trouble concentrating on schoolwork or conversations?",
    "How often do you isolate yourself from friends or classmates?",
    "Do you struggle to get out of bed in the morning?",
    "How often do you sleep too much or have trouble sleeping?"
],
'adult': [
    "How often do you feel overwhelmed by responsibilities?",
    "Do you have difficulty focusing on work or daily tasks?",
    "How often do you withdraw from friends and family?",
    "Do you find it difficult to get motivated in the morning?",
    "How often do you experience changes in your appetite or sleep?"
]
},
'emotional': {
    'Male': {
        'under18': [
            "How often do you get angry or annoyed easily at small things?",
            "Do you feel like nothing will get better in your life?",
            "How often do you feel like you're not good enough compared to others?",
            "Do you ever think about ending your life?",
            "How often have you stopped enjoying games, sports, or hobbies?"
        ],
        'adult': [
            "How frequently do you feel irritable or frustrated?",
            "Do you feel hopeless about your career or future?",
            "How often do you feel like a failure or worthless?",
            "Do you have thoughts that life isn't worth living?",
            "How often have you lost interest in your hobbies or passions?"
        ]
    },
    'Female': {
        'under18': [
            "How often do you feel sad or cry for no clear reason?",
            "Do you feel like things will never improve?",
            "How often do you feel insecure or like you don't matter?",
            "Do you have thoughts about harming yourself?",
            "How often have you lost interest in things that used to make you happy?"
        ],
        'adult': [
            "How frequently do you feel overwhelmed by emotions or cry easily?",
            "Do you feel hopeless about your future or relationships?",
```

```

        "How often do you feel guilty or like you're not good enough?",  

        "Do you have thoughts that life is not worth living?",  

        "How often have you lost interest in activities you once enjoyed?"  

    ]  

}  

}  

}  
  

# pick the right questions based on their age and gender  

if age < 18:  

    age_key = 'under18'  

else:  

    age_key = 'adult'  

if gen in ['Male', 'Female']:  

    gen_key = gen  

else:  

    gen_key = 'Male'  
  

# return both behavioral and emotional questions  

return qs['behavioral'][gen_key][age_key], qs['emotional'][gen_key][age_key]  
  

def do_assessment(gen, age):  

    # run through all the questions and collect answers  

    beh_qs, emo_qs = get_qs(gen, age) # get the right questions for this person  
  

    # part 2: behavioral questions (what they do)  

    print("PART 2: How You Act")  

    print("-" * 40)  

    print("Answer based on what you actually do.\n")  
  

    beh_scores = [] # empty list to store behavioral scores  

    for i in range(len(beh_qs)):  

        q = beh_qs[i]  

        beh_scores.append(ask_q(q, i + 1)) # ask question and add score to list  
  

    print("\n" + "=" * 60 + "\n") # separator between parts  
  

    # part 3: emotional questions (how they feel)  

    print("PART 3: How You Feel")  

    print("-" * 40)  

    print("Be honest about your feelings.\n")  
  

    emo_scores = [] # empty list to store emotional scores  

    for i in range(len(emo_qs)):
```

```

q = emo_qs[i]
emo_scores.append(ask_q(q, i + 6)) # ask question and add score to list

print("\n" + "=" * 60 + "\n") # end of assessment

return beh_scores, emo_scores # send back both lists of scores

def calc_score(beh, emo, dur):
    # calculate the final score and severity level
    tot = 0
    for b in beh:
        tot += b
    for e in emo:
        tot += e

    # if they've felt this way for a while, make the score a bit higher
    if dur in ['1 to 3 months', 'More than 3 months']:
        mult = 1.1
    else:
        mult = 1.0
    adj = tot * mult # apply the multiplier

    # figure out what level of severity based on the score
    if adj < 13:
        lvl = "Minimal"
        sev = 1
    elif adj < 20:
        lvl = "Mild"
        sev = 2
    elif adj < 28:
        lvl = "Moderate"
        sev = 3
    else:
        lvl = "Severe"
        sev = 4

    return tot, lvl, sev # send back total score, level name, and severity number

def show_results(name, cat, dur, prev, tot, lvl, sev):
    # display all the results in a nice format
    print("=" * 60)
    print(" " * 22 + "RESULTS") # centered header
    print("=" * 60)
    print("\nName: " + name) # show their name

```

```

print("Category: " + cat) # show their category
print("How long: " + dur) # show duration
print("Talked to someone before: " + prev) # show if they had previous treatment
print("\nYour Score: " + str(tot) + "/40") # show their total score out of 40
print("Level: " + lvl.upper()) # show severity level in caps
print("-" * 60)

# explain what their score means in simple terms
if sev == 1:
    msg = "You're doing okay. Everyone has bad days, and it looks like\nyou're going through normal ups and downs. Just keep an eye\non how you're feeling and take care of yourself."
elif sev == 2:
    msg = "You're having some rough patches. Things aren't great but\nthey're not terrible either. Maybe try some self-care stuff\nand see if things get better. If not, consider talking to someone."
elif sev == 3:
    msg = "Okay, so things are pretty rough right now. This is affecting\nyour daily life and you shouldn't have to deal with this alone.\nSeriously consider talking to a professional. There's no shame\nin getting help - that's what they're there for."
else:
    msg = "Listen, things are really bad right now. You need to talk to\nsomeone ASAP. Like, today if possible. You don't deserve to feel\nthis way and there are people who can actually help. Please don't\nbrush this off - reach out to someone."

print("\nWhat This Means:")
print("\n" + msg) # show the message that matches their severity
print("-" * 60)

# give them action steps based on severity
print("\nWhat You Should Do:")
if sev <= 2:
    tips = [
        "Exercise, even just walking around",
        "Eat decent food and sleep enough",
        "Talk to friends or family",
        "Do things you enjoy (or used to enjoy)",
        "Try meditation or journaling",
        "If things get worse, get professional help"
    ]
else:
    tips = [
        "TALK TO A PROFESSIONAL - like actually do it",
        "See a doctor or therapist",
        "Tell someone you trust what's going on",
    ]

```

```

    "Look into therapy and maybe medication",
    "Don't isolate yourself"
]

for t in tips:
    print("\n- " + t)

# show crisis hotline numbers
print("\n" + "=" * 60)
print(" " * 18 + "GET HELP HERE") # centered
print("=" * 60)
print("\nLUZON HOTLINES:")
print(" " * 60)
print("NCMH Crisis Hotline")
print(" 1553 (Nationwide landline toll-free) 1800-1888-1553")
print()
print("DOH Mental Health Hotline")
print(" 1555 (works on PLDT, Smart, Sun, TNT)")
print()
print("Emergency: 911")
print("-" * 60)

# extra warning for severe cases
if sev == 4:
    print("\n!! If you're thinking about hurting yourself, call one of")
    print("these numbers NOW or go to the ER. Seriously. Don't wait.")

# disclaimer and encouragement
print("\n" + "=" * 60)
print("\nBtw:")
print("This is just a screening tool, not a diagnosis. Only a real")
print("professional can tell you for sure what's going on. But if this")
print("test is showing red flags, take it seriously and get checked out.")
print("\nRemember:")

# encouraging reminders
rems = [
    "You're not the only one going through this",
    "People can actually help",
    "Things can get better",
    "You matter"
]
for r in rems:
    print("- " + r)

```



```
print("..")  
main()
```