```
def pass1_macroprocessor(source_code):

    mnt = {}  # Macro Name Table

    mdt = []  # Macro Definition Table

    intermediate_code = []

    macro_def = False

    macro_name = None


    for line in source_code:

        tokens = line.strip().split()

        if tokens[0] == "MACRO":

            macro_def = True

        elif macro_def and not macro_name:

            macro_name = tokens[0]

            mnt[macro_name] = len(mdt)

        elif macro_def and tokens[0] == "MEND":

            macro_def = False

            macro_name = None

        elif macro_def:

            mdt.append(line.strip())

        else:

            intermediate_code.append(line.strip())


    return mnt, mdt, intermediate_code


# Example usage

source_code = [

    "MACRO", "INCR &ARG", "ADD &ARG, 1", "MEND",
    "START", "MOV A, B", "INCR A", "END"
]
mnt, mdt, intermediate_code = pass1_macroprocessor(source_code)
print("MNT:", mnt)
print("MDT:", mdt)
print("Intermediate Code:", intermediate_code)


def pass2_macroprocessor(mnt, mdt, intermediate_code):
    expanded_code = []
    for line in intermediate_code:
        tokens = line.split()
        if tokens[0] in mnt:
            macro_start = mnt[tokens[0]]
            for macro_line in mdt[macro_start:]:
                expanded_code.append(macro_line.replace("&ARG", tokens[1]))
        else:
            expanded_code.append(line)
    return expanded_code


# Example usage for Pass-II
expanded_code = pass2_macroprocessor(mnt, mdt, intermediate_code)
print("Expanded Code:", expanded_code)
```

OUTPUT:

```
MNT: {'INCR': 0}

MDT: ['ADD &ARG, 1']

Intermediate Code: ['START', 'MOV A, B', 'INCR A', 'END']

Expanded Code: ['START', 'MOV A, B', 'ADD A, 1', 'END']
```