* The first of our “helper” methods will read and store an array of 10 records representing students; each record consists of a name of type string of up to 40 characters, age of type integer, and an ID of type integer.

Set up String array with 10 index

Set up Integer array with 10 index

Set up Integer array with 10 index

1. Accept Name ✅

Initialize array address

Calculate array address with index

Store name

1. Accept Age　✅

Initialize array address

Calculate array address with index

Store Age

1. Accept ID　✅

Initialize array address

Calculate array address with index

Store ID

1. Fill up Array

Loop

Ask user name

Call Accept Name

Ask user age

Call Accept Age

Ask user ID

Call Accept ID

Increment counter

Check counter is 10

* The second of our “helper” methods prints the array of records created by the first method in a readable format (a single line for each student).

1. Print Row and Name ✅

Print “Recorder”

Print current index

Access ArrayName with index

Store value

Print value

1. Print Age　✅

Access ArrayAge with index

Store value

Print value

1. Print ID　✅

Access ArrayID with index

Store value

Print value

1. Print recorder

Set index = 1

loop

Call PrintRowAndName

Call Print Age

Call print ID

check index == 11, then exit loop,

* The third “helper” method will swap any two records, and then prints the entire array.

1. Swap NameArray

Access NameArray the first user index

Store value to temp

Access NameArray the second user index

Replace value as NameArray[first user index] = NameArray[second index]

Replace value as NameArray[Second index] = temp

1. Swap AgeArray

Access AgeArray the first user index

Store value to temp

Access AgeArray the second user index

Replace value as AgeArray[first user index] = AgeArray[second index]

Replace value as AgeArray[Second index] = temp

1. Swap IDArray

Access IDArray the first user index

Store value to temp

Access IDArray the second user index

Replace value as IDArray[first user index] = IDArray[second index]

Replace value as IDArray[Second index] = temp

1. Swap values

Call SwapNameArray

Call SwapAgeArray

Call SwapIDArray

The program must ask the user to enter the record numbers. (You may assume the records are numbered from 1 to 10. For example, if the user enters 4 and 7, then the program swaps record 4 with record 7.) The main method (where your program should start), will then use these methods to create our array of records, print the records for our user to see, and then provide a menu for them to swap records or exit.

* Main

Call FillUpArray

Call PrintRecorder

Loop

Display menu

Prompt “Please choose one of the above options: ”

If user choose 2

Exit loop

If else user choose 1

Accept integer,

Prompt “Which record do you select first? ”

Accept integer as first user integer

Accept integer as the second user integer

Call swapValues

Call printRecorder

Back to loop

Else back to loop

Exit program

# Register List

# $t0 walker Index, counter

# $t1 loading address

# $t2 pointer, copy the starting address and calculate to access the value

# $t3 values from array

# $t4 Calculation actual first index = user input -1

# $t5 Calculate actual second index = user input -1

# $t6 Second pointer

# $t7 temporary value to prepare for swap

# $t8 display index: make sure array index = user input -1

# $t9 Checking condition in menu

# $s0 user input

# $s1 user menu choice

# $s2 user first index

# $s3 use second index

# $s4

# $s5

# $s6

# $s7