Spring 2025, MIS 102 – COMPUTER PROGRAMMING

Quiz 5

姓名:_____ 學號:_____ 系級:_____

1. [60 pts]

1. What does the * operator do when used with a pointer in C?
(a) Multiplies two values.
(b) Declares a pointer.
(c) Finds the address of a variable.
(d) Accesses the value at the memory address.

2. What is the character constant representing the null character in C?
(a) NULL
(b) 0
(c) '\0'
(d) False

3. Which of the following declarations means “constant pointer to constant data”?
(a) const int *ptr;
(b) int *const ptr;
(c) const int *const ptr;
(d) int const ptr;

4. Which function signature ensures a string will NOT be modified inside the function?
(a) void print(char *str).
(b) void print(const char *str).
(c) void print(char str[]).
(d) void print(char const str[]).

5. What is required to dereference a void * pointer?
(a) Assigning it to a regular pointer variable
(b) Casting to a specific type
(c) Using the & operator
(d) sizeof operator

6. If we want to define two pointers s and t. Which of the following definitions of pointers is CORRECT?
(a) int *s, *t;
(b) int *s, t;
(c) int* *s, *t;
(d) int* s, t ;

7. A C pointer is a variable that stores _____.
(a) String
(b) Floating point value
(c) Memory location

(d) True value

8. Which of the following statements about slicing in NumPy is TRUE?

(a) All slices create copies
(b) Slices always trigger a full data allocation
(c) Slices are views unless the array is non-contiguous
(d) Slices trigger garbage collection of the original array

9. If array name ary is passed to a function, C automatically passes _____.
(a) &ary[0]
(b) ary [1]
(c) ary [0]
(d) *ary

10. Consider a = np.arange(10)[::2]; what is the stride of a in memory?

(a) 1
(b) 2 * itemsize
(c) itemsize / 2
(d) Undefined

11. Which of the following values is different from the others?

```
int *Ptr;
```

(a) *Ptr
(b) *&Ptr
(c) &*Ptr
(d) Ptr

12. If you want to make a variable value modifiable but its address unmodifiable, which of the following declarations is CORRECT?

(a) const int *ptr
(b) int *const ptr
(c) const int *const ptr
(d) It is impossible to do this declaration.

13.In SciPy's csr_matrix, what happens if you assign to a specific element like m[1, 2] = 5?

(a) It updates the element in-place
(b) It raises an error
(c) It silently drops duplicates
(d) It may reallocate internal buffers

14. Which Python function provides similar information to the memory address a pointer holds in C?
(A) type()
(B) id()
(C) hex()
(D) ref()

15. Which of the following Python data types behaves most like a const int *ptr in C (pointer to constant data)?
(A) list
(B) tuple
(C) dict
(D) set

2. [40 pts]

Q1 (20 pts): Write a C recursive function reverseIntArray(intArray, out, a_size) that reverses an integer array intArray given its size a_size, and then stores the result into another array out. For example, your function, given an integer array as below,

intArray	**35**	**23**	**17**	**2**	**5**	**11**	**20**	**1**

must be able to reverse the array and save the result to another array out as:

out	**1**	**20**	**11**	**5**	**2**	**17**	**23**	**35**

- -

Requirements:

- You must define the recursive function reverseIntArray(intArray, out, a_size). (Please determine the appropriate parameter types on your own.)
- Use the array:
  ```
  int intArray[] = {35, 23, 17, 2, 5, 11, 20, 1};
  ```
  in your program to test and demonstrate that your function works correctly.
- You must use recursion — for, while, and do-while loops are not allowed.
- Your recursive logic should use only pointer arithmetic and index expressions.
- Your program must print the reversed array as the final output.
- Example output:
  ```
  Reversed array:
  1 20 11 5 2 17 23 35
  ```

Q2 (20 pts): Write a C program that demonstrates the use of a pointer to a pointer.
The program should read an integer value from the user and print it.
Then, using a pointer to a pointer, modify the original value to 10.
Finally, print the updated value.

———————————————————————————————————————

Requirements:
1. **Use two levels of indirection (int **pptr)**
2. Read an integer from the user and assign it to num
3. Use ****pptr** to set the value of num to **10**
4. Print the value before and after modification
5. **Do not modify num directly or through *ptr — only use **pptr**

———————————————————————————————————————

Here are some examples:

```
Example Input:
Enter a number: 7
Example Output:
Original value: 7
```

```
Value after pointer-to-pointer modification: 10
—--------------------------------------------------
```
Example Input:

```
Enter a number: 8
```
Example Output:

```
Original value: 8
Value after pointer-to-pointer modification: 10
```