Programming and Data Structures (Autumn 2024–25)

Full marks = 30 **Class Test 2** **Time = 1 hour**

Answer all. Write the answers in the blanks or boxes only.

NAME: ______________________ ROLL NO.: ________________ SEC.: ____

1. Answer the following questions. $\boxed{1 \times 10 = 10 \text{ marks}}$

i) Write the hexadecimal representation of the decimal number 2024. **Ans:** 7E8

ii) Write in decimal number system the normalized real value of the single-precision number: `10110011 11100000 00000000 00000000` **Ans:** -1.75×2^{-24}

iii) Which of the following rational numbers have exact values as `float`? **Ans:** $\frac{2024}{2048}, \frac{1023}{1024}$

$$p = \frac{1}{2024} \quad q = \frac{11}{4048} \quad r = \frac{2024}{2048} \quad s = \frac{2023}{2024} \quad t = \frac{1023}{1024}$$

iv) Which of the following sorting algorithms is in-place (i.e., requires no extra array) and has the least average-case time complexity? Also write its worst-case time complexity. **Ans:** Quick Sort, w.c.: $\mathcal{O}(n^2)$

Bubble Sort Selection Sort Merge Sort Quick Sort

v) A_1 and A_2 are two sorted arrays, containing 2^{2024} distinct elements and 4^{2024} distinct elements, respectively. Let k_1 and k_2 denote the respective number of elements of A_1 and A_2 that will be accessed in the worst case for binary search on A_1 and A_2. What will be the value of k_1/k_2? **Ans:** 0.499 to 0.501

vi) A and B are two matrices, with A having 10 rows and 9 columns, and B having 9 rows and 8 columns. How many scalar multiplications are required to multiply them? (A *scalar multiplication* refers to the multiplication of two numbers.) **Ans:** 720

vii) An array has 100 distinct elements. At most how many times is the function `partition` called in Quick Sort in the worst case? **Ans:** 99

viii) Write the output for the following piece of code. **Ans:** 81, 81

```
int n = 3, *p;
p = &n; *p = n*n; n = *p * *p;
printf("%d, %d", *p, n);
```

ix) Write the output for the following piece of code for `n` $= 47$. **Ans:** `101111`

```
void printFun(unsigned int n){
  if (n > 1) printFun(n >> 1);
  printf("%d", n & 1);
}
```

x) Write the C statement for dynamic memory allocation of an array of 100 integers using `malloc`.

Ans: `p = (int *)malloc(100*sizeof(int));`

2. The following code computes the addition of two small matrices, stores it in a new matrix, and prints its elements on the terminal. Each row is printed on a new line. Note that only one extra variable (denoted by `i`) is used throughout the code. There is no allowance for declaring or using any other variables. Fill in each blank with exactly one expression. 10 marks

```
#include<stdio.h>

int main(){
  int a[2][3], b[2][3], c[2][3], i;

  printf("Enter Matrix 1:\n");

  for(i=0; _____________________________; i++)              // 1 mark

    scanf("%d", _________________________________________); // 1 mark


  printf("Enter Matrix 2:\n");

  for(i=0; _____________________________; i++)              // 1 mark

    scanf("%d", _________________________________________); // 1 mark

  /* compute the sum */

  for(i=0; _____________________________; i++)              // 1 mark

    c[________][________] = a[________][________] + b[________][________];//2 marks


  printf("Matrix sum:\n");

  for(i=0; _____________________________; i++)              // 1 mark

    printf("%d ", _______________________________________); // 1 mark


    if (_________________________________________________)  // 1 mark

      printf("\n"); // new line for the next row
  }

  return 0;
}
```

ANSWER

```
for(i=0; i<6; i++)

&a[0][0]+i
or
(i<3 ? &a[0][i] : &a[1][i-3])

for(i=0; i<6; i++)
```

```
&b[0][0]+i
or
(i<3 ? &b[0][i] : &b[1][i-3])


for(i=0; i<6; i++)
  c[i/3][i%3] = a[i/3][i%3] + b[i/3][i%3];
  or
  c[i%2][i%3] = a[i%2][i%3] + b[i%2][i%3];
  or
  c[i<3 ? 0:1][i<3 ? i:i-3] = a[i<3? 0:1][i<3 ? i:i-3] + b[i<3 ? 0:1][i<3 ? i:i-3];

for(i=0; i<6; i++){
  printf("%d ", c[i/3][i%3]);
  or
  printf("%d ", (i<3 ? c[0][i] : c[1][i-3]));

  if ((i+1)%3 == 0)
  or
  if (i%3 == 2)
  or
  if (i == 2)
    printf("\n"); // new line for the next row
}
```

Space for rough work

3. The following code takes two complex numbers as input, multiplies them, and prints the result. For example, for the complex numbers $2+3i$ and $-1+4i$, it will print `-14.00 + 5.00i`. Fill in the blanks. $\boxed{1\times 6 = 6 \text{ marks}}$

```
#include <stdio.h>

typedef struct {
  float real, imag;
} complex;

complex multiply(complex c1, complex c2) {

  __________________________________________; // declare a single variable

  result.real = ___________________________________________________;

  result.imag = ___________________________________________________;

  return result;
}

int main() {
  complex c1, c2, result;

  printf("Enter real and imaginary part of the first complex number: ");

  scanf("%f%f", ___________________________________________________);

  printf("Enter real and imaginary part of the second complex number: ");

  scanf("%f%f", ___________________________________________________);

  result = ________________________________________________________ ;

  printf("The product of the complex numbers is: %.2f + %.2fi\n",
         result.real, result.imag);

  return 0;
}
```

ANSWER

```
complex result
c1.real * c2.real - c1.imag * c2.imag
c1.real * c2.imag + c1.imag * c2.real

&c1.real, &c1.imag
&c2.real, &c2.imag
multiply(c1, c2)
```

Space for rough work

4. What is the minimum value of `n_iter` that must be provided as input to the following code to ensure that the given array `a[]` is sorted by the time the `for`-loop in `main()` ends? 4 marks

```c
#include <stdio.h>
#define SWAP(a, b) { int temp = a; a = b; b = temp; }

void loop(int a[], int i){
  for (int j = 0; j < i; j++)
    if (a[j] > a[j + 1])
      SWAP(a[j], a[j + 1]);
}

int main() {
  int i, n_iter, a[] = {7, 3, 1, 2, 6, 8, 7, 9, 8, 9};
  scanf("%d", &n_iter);

  for(i = 0; i < n_iter; i++)
    loop(a, 9-i);

  return 0;
}
```

Ans: 2

Space for rough work