

Roll No.:

Name:

Department of CSE

IIT Kharagpur

Autumn 2025-26

Programming and Data Structures (CS10003) – CT2 Full Marks: 20

Duration: 1 Hour

Note: Your final answer written in Blue/Black Pen inside the space provided will only be evaluated.

Do not show any rough work inside those spaces.

1. Let us consider a computer, where integer variables are stored in 4 bytes and the size of any pointer is 8 bytes. Assuming the malloc call succeeds, what will get displayed when the following program is executed? [2]

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int* ptr;
    ptr = (int*) malloc(50 * sizeof(int*));
    printf("%d", sizeof(ptr));
    return 0;
}
```

8

2. What will get displayed when the following program is executed? [2]

```
#include <stdio.h>
int main() {
    int a[] = {10, 20, 30, 40, 50};
    int *p = a;
    *(p + 2) = *(a + 4) - *(p + 1);
    printf("%d", a[3]);
    return 0;
}
```

40

3. What will get displayed when the following program is executed? In case the program returns an error (Runtime or Compile time), mention ERROR as the answer. [2]

```
#include <stdio.h>
void update(int *p) {
    p[-1] += *p;
    int t = *p;
    *p = p[1] - p[-1];
    p++;
    *p += t;
    p[-1] = (*p)++ - p[-2];
}

int main() {
    int a[] = {1, 2, 3, 4};
    update(a + 1);
    printf("%d", a[2]);
    return 0;
}
```

6

4. What will get displayed when the following program is executed? [2]

```
#include <stdio.h>
int main()
{
    int a[3][4] = {{10,20,30,40},{50,60,70, 80}, {90,100,110,120}};
    printf("%d,%d",*((*a+1)+2), *((*a+1)+2));
    return 0;
}
```

40,22

5. What will get displayed when the following program is executed? [2]

```
#include<stdio.h>
int *func(int x, int *y) {
    x = 5;
    *y = 7;
    return y;
}

int main() {
    int a = 3, b = 5, c = 0;
    c = *func(a, &b);
    printf("%d %d %d", a, b, c);
    return 0;
}
```

3 7 7

6. Consider the following variant of merge sort called Abs-Stable Merge Sort that sorts numbers based on their absolute values in ascending order. However, when two numbers have the same absolute value, the negative one should appear before the positive one. Also if both numbers are identical, their original order should be preserved to maintain stability.

For example, consider the following input: -3 1 -1 3 -2 2 0 3

Sorting by absolute values in ascending order gives the order of magnitudes: 0, 1, 1, 2, 2, 3, 3, 3. For equal absolute values, negatives come before positives, and identical numbers retain their input order. Hence the output will be: 0 -1 1 -2 2 -3 3 3

Fill up the blanks in the following C program to read an integer array (with maximum 100 elements) from input, perform the abs-based stable sort, and print the sorted array as space-separated integers. [1×10=10]

[The code is shown in the next page.]

```

#include <stdio.h>
#include <stdlib.h>

int before(int x, int y) {
    int ax=abs(x), ay=abs(y);
    if(ax!=ay) return ax<ay;
    if((x<0)!=(y<0)) return x<0;
    return 0;
}

void merge(int *a, int l, int m, int r) {
    int n1=m-l+1, n2=r-m, i=0, j=0, k=l;
    int L[n1],R[n2];

    for(i=0;i<n1;i++) L[i]= a[l+i] ;

    for(j=0;j<n2;j++) R[j]= a[m+1+j] ;

    i=j=0;
    while(i<n1&& j<n2)
        a[k++]=before(L[i],R[j])? L[i++] : R[j++];

    while(i<n1) a[k++]= L[i++];

    while(j<n2) a[k++]= R[j++];

}

void msort(int *a, int l, int r) {
    if(l<r) {
        int m=(l+r)/2;

        msort(a, l, m); msort(a, m+1, r);

        merge(a, l, m, r);

    }
}

int main() {
    int n, i, a[100]; scanf("%d",&n);
    for(i=0;i<n;i++) scanf("%d",&a[i]);

    msort(a, 0, n-1);

    for(i=0;i<n;i++) printf("%d%c",a[i],i+1==n?'\\n':' ');
}

```

Space for Rough Work