University of Bordj Bou Arreridj
Faculty of Mathematics and Computer Science
Computer Science Department — L1 CS

EXAM — ASD

Arrays · Pointers · Linked Lists · Functions

Duration: 2h | No documents allowed

Group: _____
Copy #: _____

Answer only on the OCR answer sheet

Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5	Total
4 pts	4 pts	4 pts	4 pts	4 pts	20 pts

Exercise 1 — Pass by Reference vs. Pass by Value — 4 pts

Analyze the program below and answer the questions.

```c
#include <stdio.h>

void f(int *a, int b)
{
    *a = *a + b;
    b  = b + 5;
}

int main()
{
    int x = 2, y = 3;
    f(&x, y);
    printf("%d %d", x, y);
    return 0;
}
```

Value of `x` in `main` after the call to `f` ?	1 pt	[E1_Q1]
Value of `y` in `main` after the call to `f` ?	1 pt	[E1_Q2]
Value of `b` at line 6, just before `f` returns ?	1 pt	[E1_Q3]
Output produced by `printf` at line 13 ?	1 pt	[E1_Q4]

Exercise 2 — Pointers and Side Effects — 4 pts

Trace the state of variables after each instruction in function g.

```c
void g(int *p, int *q)
{
    *p = *p + 1;   /* (A) */
    p  = q;        /* (B) — p now points to b */
    *p = *p + 2;   /* (C) */
}

int main()
{
    int a = 5, b = 10;
    g(&a, &b);
    printf("%d %d", a, b);
    return 0;
}
```

After instruction **(A)**, what is the value of `a` ?	1 pt	[E2_Q1]
After instruction **(C)**, what is the value of `b` ?	1 pt	[E2_Q2]
Does instruction **(B)** modify `a` or `b` in `main` ? *(True / False)*	1 pt	[E2_Q3]
Output produced by `printf` at line 12 ?	1 pt	[E2_Q4]

Exercise 3 — Pointer Arithmetic — 4 pts

```c
void process(int *p)
{
    *(p+1) = *p + *(p+2);
}

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

Value of `T[0]` after the call ?	0.5	[E3_Q1]
Value of `T[1]` after the call ?	1 pt	[E3_Q2]
Value of `T[2]` after the call ?	0.5	[E3_Q3]
Value of `T[3]` after the call ?	0.5	[E3_Q4]
Output produced by `printf` at line 10 ?	1.5	[E3_Q5]

Exercise 4 — Debugging — 4 pts

⚠ The code below contains **3 errors**. Find them and give the correction.

```c
#include <stdio.h>

void fill(int T, int n)
{
    int i;
    for(i=0; i <= n; i++)
    {
        scanf("%d", T[i]);
    }
}
```

Error line 3 — type of parameter `T` : correction ?	1.5	[E4_Q1]
Error line 6 — loop condition : correction ?	1 pt	[E4_Q2]
Error line 8 — `scanf` call : correction ?	1.5	[E4_Q3]

Exercise 5 — Linked Lists and Nested Function Calls

4 pts

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct node {
5      int          val;
6      struct node *next;
7  } node;
8
9  int f(int x) { return x + 1; }
10
11 int g(int x) { return f(x) * 2; }
12
13 int main()
14 {
15     node *head = NULL;
16     node *n1 = malloc(sizeof(node));
17     node *n2 = malloc(sizeof(node));
18     node *n3 = malloc(sizeof(node));
19     n1->val=10; n2->val=20; n3->val=30;
20     head=n1; n1->next=n2; n2->next=n3;
21     n3->next=NULL;
22     printf("%d %d %d",
23       head->val,
24       head->next->val,
25       head->next->next->val);
26     printf(" %d", g(3));
27     return 0;
28 }
```

Analyze the linked list and the nested function calls:

Value printed at line 23 — `head→val` ?	0.5	[E5_Q1]
Value printed at line 24 — `head→next→val` ?	0.5	[E5_Q2]
Value printed at line 25 — `head→next→next→val` ?	0.5	[E5_Q3]
Value returned by `f(3)` — line 9 ?	0.5	[E5_Q4]
Value returned by `g(3)` — line 26 ?	2 pt	[E5_Q5]

Good luck! · Bonne chance! · بالتوفيق

University of Bordj Bou Arreridj
Faculty of MI — L1 Computer Science

ANSWER SHEET — ASD

ورقة الإجابة — Automated OCR Processing

Group: _____
Copy #: _____

⚠ Important Instructions — تعليمات هامة

- ✓ Write in black ink, clearly, within the box
- ✓ No correction fluid or crossing out allowed
- ✓ Any violation will result in the answer being discarded

- ✓ الكتابة بحبر أسود واضح داخل الإطار فقط
- ✓ ممنوع استعمال المبيض أو الشطب أو الكتابة فوق الإجابة
- ✓ أي مخالفة تؤدي إلى عدم احتساب الإجابة تلقائياً

Question ID	Answer / الإجابة	✓ Expected Answer	Pts
Exercise 1 — Pass by Reference vs. Pass by Value			
[E1_Q1]		5	1 pt
[E1_Q2]		3	1 pt
[E1_Q3]		8	1 pt
[E1_Q4]		5 3	1 pt
Exercise 2 — Pointers and Side Effects			
[E2_Q1]		6	1 pt
[E2_Q2]		12	1 pt
[E2_Q3]		False — only local pointer p changes	1 pt
[E2_Q4]		6 12	1 pt
Exercise 3 — Pointer Arithmetic			
[E3_Q1]		1	0.5 pt
[E3_Q2]		4	1 pt
[E3_Q3]		3	0.5 pt
[E3_Q4]		4	0.5 pt
[E3_Q5]		1 4 3 4	1.5 pt
Exercise 4 — Debugging			
[E4_Q1]		int *T (or int T[])	1.5 pt
[E4_Q2]		i < n	1 pt
[E4_Q3]		scanf("%d", &T[i])	1.5 pt
Exercise 5 — Linked Lists and Nested Function Calls			
[E5_Q1]		10	0.5 pt
[E5_Q2]		20	0.5 pt
[E5_Q3]		30	0.5 pt
[E5_Q4]		4	0.5 pt
[E5_Q5]		8 (f(3)=4 → 4×2=8)	2 pt