

UNIVERSITY OF KWAZULU-NATAL
School of Engineering
Discipline of Electrical, Electronic and Computer Engineering
(Howard College Campus)

Main Examinations: **May 2026**

ENEL2CA H1 : Computer Methods 1

DURATION :2 HOURS

MARKS : 100

Examiners:

Prof. J.R. Tapamo

Mr. J. W. Parker

Instructions:

1. Answer ALL questions.
2. Ensure that your answers are clearly numbered.
3. Although 106 marks are available, this test will be marked out of 100 marks.
4. Calculators are not allowed.
5. Answer **Section A** on the multiple choice answer sheet provided.
6. Ensure you have 11 pages, including this one.
7. **This paper may not be removed from the exam hall.**

Section A: Multiple Choice**[38 marks]**

Answer on the multiple-choice sheet. Use an HB pencil only. For each question choose the best of the available answers.

1. Which declaration correctly declares a pointer to an integer?

- (A) int ptr;
- (B) pointer int ptr;
- (C) int *ptr;
- (D) *int ptr;
- (E) ptr int*;

[1]

2. Which declaration correctly creates an array of 10 integers?

- (A) int arr;
- (B) array int arr[10];
- (C) int arr(10);
- (D) int arr[10];
- (E) arr int[10];

[1]

3. After the execution of the following fragment of code

```
int a = 3, b = 6;  
int x = --b * (b++ - a) + b / (--a);
```

what will the variables `x` `a` `b` hold?

- (A) `x = 13, a = 2, b = 6`
- (B) `x = 15, a = 2, b = 6`
- (C) `x = 13, a = 2, b = 5`
- (D) `x = 15, a = 3, c = 5,`
- (E) `x = 13, a = 3, c = 6`

[2]

4. Given the following definitions, what is the value of `Array[Array[2][0]][Array[1][1]]`?

```
int Array[4][5] = {{2, 11}, {-1, 3}, {3, 2, 5}, {4, -2, 6, 8, 11}};
```

- (A) 8
- (B) 11
- (C) 5
- (D) 11
- (E) 0

[2]

5. Given the following fragment of code

```
#include <stdio.h>
int main()
{
    int a = 1, b = 1;
    switch (a)
    {
        case a*b:
            printf("yes ");
        case a-b:
            printf("no\n");
            break;
        case a+b:
            printf("ho\n");
            break;
    }
}
```

What will be the result of its execution?

- (A) yes
- (B) no
- (C) compile error
- (D) yes no
- (E) ho

[2]

6. What is output after the execution of the following C code fragment?

```
char str[] = "ABC";
str[0] = 'X';
printf("%s", str);
```

- (A) ABC
- (B) XBC
- (C) error
- (D) Garbage value
- (E) None of the above

[2]

7. Which of the following statements is true about local variables?

- (A) Accessible globally
- (B) Stored permanently
- (C) Accessible only inside the function
- (D) Initialized to zero automatically
- (E) None of the above

[1]

8. Which of the following statements is true about `switch`?

- (A) It supports floating-point cases
- (B) Duplicate case values are allowed
- (C) Default case is mandatory
- (D) Case labels must be constant expressions
- (E) None of the above

[1]

9. Which statement about multidimensional arrays is true?

- (A) They cannot store integers
- (B) They are arrays of arrays
- (C) They require pointers only
- (D) They are illegal in C
- (E) None of the above

[1]

10. What is the results of execution of the following fragment of C code.

```
int i = 1;
while(i < 5)
{
    if(i == 3)
        break;
    printf("%d ", i);
    i++;
}
```

- (A) 1 2
- (B) 1 2 3
- (C) 1 2 3 4
- (D) infinite loop
- (E) None of the above

[2]

11. Which of the following statements correctly checks end-of-file?

- (A) `if(file == EOF)`
- (B) `if(fp == END)`
- (C) `if(endfile(fp))`
- (D) `if(feof(fp))`
- (E) None of the above

[1]

12. Assume that the file "**numfile.txt**" contains the following numbers:

1 2 3 4

What is printed out after the execution of the following C code fragment?

```
#include <stdio.h>

int main()
{
    FILE *fp;
    int n;

    fp = fopen("numfile.txt", "r");

    while(fscanf(fp, "%d", &n) != EOF)
    {
        printf("%d ", (n%2)? n * n:n);
    }
    fclose(fp);
    return 0;
}
```

- (A) 1 2 3 4
- (B) 1 2 9 4
- (C) 2 4 6 8
- (D) 1 4 9 16
- (E) 1 2 9 16

[3]

13. What value does function `compute` return when called with a value 29531:

```
int compute(int x)
{
    if (x > 0)
        return ((x%10)%3)?(x%10):0+compute(x/10);
    return 0;
}
```

- (A) 8
- (B) 11
- (C) 2
- (D) 20
- (E) 9

[3]

14. What happens if a recursive function has no base condition?

- (A) he program executes normally
- (B) The recursion stops automatically
- (C) The compiler fixes the error
- (D) Infinite recursion may occur
- (E) The function becomes iterative

[1]

15. Consider the main function below. A function `rearrange` is called to rearrange values in three variables `num1`, `num2`, and `num3`.

```
#include <stdio.h>
void rearrange(int a, int b,int *c);
main()
{
    int num1 = 23, num2 = -24, num3 =10;
    rearrange(num1, num2,&num3);
    printf("\num1 = %d , num2 = %d , num3 = %d;", num1, num2,num3);
}

void rearrange(int a, int b,int *c)
{
    int a1;
    a1 = a>b?a:b;
    *c = a1<*c?*c:a1;
}
```

After the calling of function **rearrange** Which of the following is the correct?

- (A) num1 = 23 , num2 = -24, num3 = 23;
- (B) num1 = -24 , num2 = 10, num3 = 23;
- (C) num1 = -24 , num2 = 23, num3 = 23;
- (C) num1 = 10 , num2 = 23, num3 = -24;
- (D) num1 = 10 , num2 = 23, num3 = 23;

[3]

16. What is output after the execution of the following C code fragment?

```
int x = 5;
int *p = &x;
*p = 10;
printf("%d", x);
```

- (A) 5
- (B) 10
- (C) 7
- (D) Address of x
- (E) error

[2]

17. Given the following C code:

```
#include <stdio.h>
#include <string.h>

struct Item
{
    char name[20];
    int quantity;
};

int main()
{
    struct Item items[2] =
    {
        {"pen", 3}, {"book", 5}
    };
    int total = 0;

    for (int i = 0; i < 2; i++)
    {
        total += items[i].quantity * strlen(items[i].name);
    }
    printf("%d", total);
    return 0;
}
```

Execution of the above code will print the following number?

- (A) 14
- (B) 20
- (C) 29
- (D) 32
- (E) None of the above

[2]

18. What is printed out after the execution of the following C code fragment?

```
#include <stdio.h>
void change()
{
    static int x = 2;
    static int y = 1;
    x++; y++;
    printf( "%d -- %d  ", x, y );
}
int main()
{
    change(); change(); change();
    return 0;
}
```

- (A) 2 – 3 3 – 4 4 – 5
- (B) 2 – 1 3 – 2 4 – 3
- (C) 2 – 1 2 – 2 2 – 3
- (D) 1 – 4 3 – 4 3 – 3
- (E) 3 – 2 4 – 3 5 – 4

[3]

19. Given the following declarations and instructions:

```
typedef struct
{
    char   RoadID[6];
    char   RoadName[15];
    int    Length;
} ROAD;
ROAD  roads[] ={ {"PKZN02", "QueenNandi", 800},
                 {"UCPT01", "BNaude", 200}
};
ROAD *p;
p = &roads[0];
```

Which of the following instructions is valid?

- (A) `printf("\nThe correct ID is = %s",p.RoadID);`
- (B) `printf("\nThe Road length is = %d", (*p).Length);`
- (C) `printf("\nThe second character road ID Name is = %c", p.RoadID[1]);`
- (D) `printf("\nThe Length of item 1 = %d",p.Length);`
- (E) `printf("\nThe Length of item 1 = %d", (*p)->Length);`

[2]

20. Given the following fragment of code

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

typedef struct Node {
    int data;
    struct Node* next;
}Node;

int main() {
    Node *n1, *n2, *n3;
    n1=malloc(sizeof(Node));n2=malloc(sizeof(Node));n3=malloc(sizeof(Node));
    n1->data = -15; n2->data = 10; n3->data = 21;
    n1->next = n2; n2->next = n3; n3->next = n1;

    Node* ptr = n1;

    ptr = ptr->next; ptr = ptr->next;ptr=ptr->next;

    printf("%d", ptr->data);
    return 0;
}
```

Execution of the above code will produce the following result?

- (A) -15
- (B) 10
- (C) 21
- (D) NULL
- (E) none of the above

[2]

Section B: Program execution, testing, and errors correction**[22 marks]**

21. In this question codes are checked to identify and correct errors, if there are any:

- (a) Is the following `if` statement legal in C?

```
int n;
if (n == 1-6)
    printf("n is between 1 and 10");
```

If so, what does it do when `n` is equal to 6?

[2]

- (b) The following function is supposed to return 1 if any element of the array `table` has the value 0 and 0 if all elements are nonzero. Unfortunately, it contains an error. Find the error and show how to correct it.

[7]

```
int has_zero(int table[] ,int n)
{
    int i;
    for(i=0;i<n;i++)
        if (table[i] == 0 )
            return 1;
    else return 0;
}
```

- (c) The following program is suppose to replace all occurrence of character `o` in the string `x` by a character `d`. Correct this code. Change as little code as possible.

[7]

```
// replace occurrence of character o by character d
// in string x
void replace(char o, char d, char *x)
{
    char *t = x[];
    int i =0;
    for(; t !='\0';t--)
    {
        if (*t == o) t[i] = d;
    }
}
```

22. Let `f` be the following function:

[6]

```
#include <stdio.h>
int f(char *s, char *t)
{
    char *p1, *p2;
    for (p1=s; *p1; p1++)
    {
        for(p2 = t; *p2; p2++)
            if(*p1 == *p2) break;
        if(*p2 == '\0') break;
    }
    return p1 - s;
}
```

- (a) What is the value of `f("abcd", "babc")`
 (b) What is the value of `f("abcd", "bcd")`

Section C: Code completion and equivalent codes**[12 marks]**

23. Give an equivalent code to the code below, where all pointers are replaced by subscripts and the **while** loop is replaced by a **for** loop? [6]

```
int count_odd(int a[], int n)
{
    int nbr_odd = 0;
    int *p = a, *pEnd=p+n;
    while(p < pEnd)
    {
        if (*p%2) nbr_odd++;
        p++;
    }
    return nbr_odd;
}
```

24. Consider the definition of a simply linked list of integers in which the individual elements have the following type:

```
typedef struct node {
    int    data;
    struct node *Link;
} IntList;
```

With the following declaration:

```
IntList *List = NULL;
```

Assume the list in *Figure 2* below is dynamically created and that `List` contains the address of the first node. The fragment of C code in *Figure 1* is supposed to insert -7 as the first node of the linked list, but there are missing instructions.

- (a) What is missing at \$\$\$ 1 \$\$\$? [2]
 (b) What is missing at \$\$\$ 2 \$\$\$? [2]
 (c) What is missing at \$\$\$ 3 \$\$\$? [2]

```
IntList *p1,*p2 ; /* pointers */
p1 = malloc(sizeof(IntList));
if(p1 != NULL)
{
    /* Insert -7as value of data field of p1 */
    $$$ 1 $$$
    /* Link p1 with the current first node of the list*/
    $$$ 2 $$$
    /* Make p1 the first element of the list */
    $$$ 3 $$$
}
```

Figure 1: Code to insert -7 at the beginning of the list shown in *Figure 2*

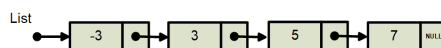


Figure 2: **Linked-List made of -3,3, 5, and 7**

Section D: Design and Coding**[34 marks]**

25. The following sum

$$f(n) = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} + \dots + (-1)^n \frac{1}{(2n+1)} + \dots$$

tends to $\frac{\pi}{4}$ as n tends to ∞ (infinity). Write a function that uses $f(n)$ to compute π with a precision of 10^{-4} [10]

26. You have been tasked to design a program that will be embedded in the control system of a self-scan checkout machine. You particularly have to design a subsystem that issues the change based on a purchase price and the available change in the machine. Suppose the money is measured in integer numbers of cents. For the South African Rand, the possible type of changes include bills (in units of 20000, 10000, 5000, 2000, and 1000 cents) and coins (in units of 500, 200, 100, 50, 20, and 10 cents).

(a) The subsystem to be designed must have a data structure that stores the values and quantities of bills and coins in the machine. Define a new C type `ChangeQtyAvail` to represent this data structure. [6]

(b) Define a C function, `MakeChange`, that takes as arguments:

- the state of availability of coins and bills in the machine (`StateOfAvail`)
- the purchase price (`PurchasePrice`)
- the money paid (`MoneyPaid`)
- the array that will contain the change (`change`)

and returns a structure reporting the number of each type of change paid out (meaning, how many 20000 cents, how many 10000 cents, and so on). The `Makechange` function should also decrement the store of change according to the change paid out. In determining change, large denomination are preferred over smaller denomination, and bills are preferred over coins. The function will then return 1 if the operation is successful, and will return -1 if there is not enough bills and coins to make the change. For instance, if the state of availability of coin in the machine is the following

Value of bills or coins	20000	10000	5000	2000	1000	500	200	100	50	20	10
Quantity of bills or coins	0	4	10	25	40	26	10	30	55	12	4

and you have given 10000 cents to pay for an item that costs 9510 cents. The change will then be 490 cents, and your function must generate the following change array

{0,0,0,0,0,0,2,1,2,0}

and return 1; meaning that the change will be made of the following:

- 2 coins of 200 cents
- 1 coins of 50 cents
- 2 coins of 20 cents

[18]