

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

Examinations: **May 2014**

ENEL2CA H1 : Computer Methods 1

DURATION : 2 HOURS

MARKS : 100

Examiners:

Prof. J.R. Tapamo

Mr. B. Naidoo

Instructions:

1. Answer ALL questions.
2. Ensure that your answers are clearly numbered.
3. Calculators are not allowed.
4. Answer **Section A** on the multiple choice answer sheet provided.
5. Ensure you have 9 pages, including this one.

Section A: Multiple Choice**[36 marks]**

Answer on the multiple-choice sheet. Use an HB pencil only. Negative marking will be used. Correct (+3), not attempted (0), incorrect (-0.75). For each question choose the best of the available answers.

1. The definition of the statements in which an identifier is known (or can be used) is its

- (A) global
- (B) local
- (C) scope
- (D) static
- (E) None of the above

2. After the execution of the following fragment of code

```
int x = 4;
int y = 2;
char op = '*';
switch (op)
{
    default : x += 3;
    case '+' : x += y; break;
    case '-' : x -= y;
}
```

what will the variable x holds?

- (A) 4
- (B) 6
- (C) 5
- (D) 9
- (E) 7

3. Assume that a and b are pointers to integers and that a points to the object $name$. What is the effect of the statement

$$a = b$$

- (A) The value of $name$ is copied into b .
- (B) The memory address stored in a is copied into b .
- (C) The memory address stored in b is copied into a .
- (D) The pointer a is now pointing to a different object.
- (E) None of the above.

4. What value does function `evaluate` return when called with a value 4:

```
int evaluate(int n)
{
    if(n == 0 ) return 4;
    else return evaluate(n-1) + evaluate(n-1);
}
```

- (A) 32
- (B) 8
- (C) 64
- (D) 16
- (E) none of the above

5. Consider the main function below. A function `Mode` is called to calculate the mode of the histogram represented by the array **Image**.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
Mode(int Image[], int dim, int hist[], int *mode);

int main() {
    int Image[] = {1,2,3,3,3,3,2,1,1};
    int hist[] = {0,0,0,0};
    int dimension = 9;
    int i, mode = 0;
    Mode(Image, dimension, hist, &mode);
    printf("\nmode = %d",mode);
    for(i=0; i < 4; i++) printf(", hist[%d] = %d ",i, hist[i]);
    system("pause");
    return 0;
}

Mode(int Image[], int dim, int hist[], int *mode)
{
    int i;
    for(i=0; i<dim; i++)
    {
        hist[Image[i]]++;
    }
    for (i=1;i<4;i++) if (hist[*mode]<hist[i]) *mode =i;
}
```

what will be printed after the execution of the above code?

- (A) mode = 3, hist[0] = 0 , hist[1] = 3 , hist[2] = 2 , hist[3] = 4;
- (B) mode = 0, hist[0] = 0 , hist[1] = 0 , hist[2] = 0 , hist[3] = 0;
- (C) mode = 4, hist[0] = 0 , hist[1] = 3 , hist[2] = 2 , hist[3] = 4;
- (D) mode = 3, hist[0] = 0 , hist[1] = 1 , hist[2] = 1 , hist[3] = 3;
- (E) None of the above;

6. Given the following fragment of code

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
char *Extract(char *ptr)
{
    ptr +=3;
    return (ptr);
}
int main()
{
    char *x, *y;
    x = "SERVICE";
    y = Extract(x);
    printf("l = %d\n", strlen(y));
    system("pause");
    return 0;
}
```

what will be printed after the execution of the above code(`strlen(s)` returns the length of the string `s`)?

- (A) l=5
- (B) l=4
- (C) l=7
- (D) l=3
- (E) l=2

7. Assume that the file "**number.txt**" contains the following numbers:

23 13 33 23 43 10 23 33 13 10

What is printed out by the following C code fragment?

```
FILE *infile;
int i, j, sum = 0;
infile = fopen("number.txt", "r");
while (fscanf(infile, "%d %d", &i, &j) != EOF)
    sum += i < j? i: j%2;
printf("%i", sum);
```

- (A) 36
- (B) 79
- (C) 55
- (D) 98
- (E) 135

8. 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;
```

Given the following fragment of code

```
/* Insert a new element in a list */
void Insert(IntList **ptr, float item)
{
    IntList *p2; /* pointer that will enable us to create a new node */
    p2 = malloc(sizeof(IntList));
    if(p2 !=NULL)
    {
        ???
    }
}
```

Which one of the following can replace ??? for the function call **Insert(List, item)** to insert a node with a data part containing item at the beginning of **List**?

- (A) `p2->data = item; if(*ptr==NULL) {*ptr = p2; p2->Link = NULL;}`
`else {p2->Link = *ptr; *ptr = p2;}`
- (B) `p2->data = item; if(*ptr==NULL) {*ptr = p2;}` `else {p2->Link = *ptr; *ptr = p2;}`
- (C) `p2->data = item; if(*ptr==NULL) {*ptr = p2; p2->Link = NULL;}`
`else {p2->Link = *ptr;}`
- (D) `if(*ptr==NULL) {p2->Link = NULL;}` `else {p2->Link = *ptr; *ptr = p2;}`
- (E) None of the above

9. Given the following declarations

```
int i[] = {11, 20, 67, 10};
int *pa[] = {i+2, i+3, i, i+1};
```

What is the value of `*pa[2]`?

- (A) 20
- (B) 11
- (C) 67
- (D) 10
- (E) none of the above

10. Given the following declarations:

```
typedef struct
{
    char Name[10];
    int Size;
} City;
typedef struct
{
    char Name[10]; City Cities[2];
} Country;

Country Regions[3]
={
    {"ClearSkies",{{"Blue",1000},{"limpid", 10000}}},
    {"Welcome",{{"Rainy",100}, {"Happytown", 200}}},
    {"NiceRegion", {{"SandyBeach", 100},{"SunnyTown", 10000}}}
};
Country *ptr = &Regions[0] ;
```

What will be the result of the execution of the following fragment of code?

```
printf("%s, ", (ptr)->Name); printf("%s, ", (ptr+1)->Cities[1].Name);
printf("%d", ++ptr->Cities[1].Size);
```

- (A) ClearSkies, Happytown, 200
- (B) NiceRegion, SandyBeach, 10000
- (C) Clearskies, Limpid, 200
- (D) Welcome, Happytown, 1000
- (E) none of the above

11. Which can be used to determine the size of an array `a [ ]`?

- (A) `sizeof(a[])/sizeof(a)`
- (B) `sizeof(a)/sizeof(a[])`
- (C) `sizeof(a)/sizeof(a[0])`
- (D) `sizeof(a[0])/sizeof(a)`
- (E) none of the above

12. What is a proper way to open the file **FileName.bin** for reading as binary file?

- (A) `FILE *f = fread("FileName.bin","b");`
 - (B) `FILE *f = fopen("FileName.bin","rb");`
 - (C) `FILE *f = fopenb("FileName.bin","w");`
 - (D) `FILE *f = fread("FileName.bin","rb");`
 - (E) `FILE *f = freadb("FileName.bin","rb");`
-

Section B: Program Execution and Testing**[32 marks]**

13. Write a single C statement to accomplish each of the following:

- (a) Declare the variables `time`, `ThisValues`, and `A370` to be of type `float`. [1]
- (b) Read three integers from the keyboard and store them in the variables `x`, `y`, and `z`. [1]

14. Identify and correct the errors in each of the following fragments of code:

- (a) Given an integer `value`, The following function, `OddEven(value)`, is supposed to return 1 if `value` is odd and 0 if `value` is even: [3]

```
void OddEven(float x)
{
    return(value/2);
}
```

- (b) The following piece of code should print all even values between 1 and 10 included. [2]

```
n = 4;
while(n <= 10)
{
    printf("%d ", n);
    n += -1;
}
```

- (c) The following piece of code should receive a real number as argument and print it. [3]

```
void f(x)
{
    float x;
    printf("%d", x);
}
```

- (d) The following fragment of code should read 10 integers and store them in `B`. [3]

```
int *B[10] , i;
for(i = 0; i < 10; i--)
{
    scanf("%d", B[i]);
}
```

- (e) The following code should convert Upper case letters of a string to lower case [3]

```
/* function that converts Upper case
   letter of the string of character
   str into lower case
*/
void UpperToLower(float str[])
{
    int i;
    for(i=0; str[i]!="\0"; i--)
        st(i) = ((str[i] >= 'A' && str[i] <= 'Z')
                ?str[i]-'A'+'a': str[i]);
}
```

15. The following source code is a C implementation of selection sorting method that sorts a given array of integers in increasing order. Fill out blanks in the code. [4]

```
void selectionSort(int a[], int N)
{
    int i, j, min, t;
    for (i=0; i<N-1; i++) {
        min=i;
        for (j=i+1; j<N; j++)
            -----
        t=a[min]; a[min]=a[i]; a[i]=t;
    }
}
```

16. What is the output (values in **b** and **p**) of the execution of the following code? Show your working. [12]

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#define SIZE 7
int Pivoting(int b[], int size);
main()
{
    int b[SIZE] = {-1,4,-8,-10,23,5,3};
    int i, p = Pivoting(b,SIZE);
    printf("\nThe Value the pivot is = %d\n", p);
    for(i=0; i<7;i++)printf("\nb[%d]=%d",i,b[i]);
    system("pause");
}

int Pivoting(int b[], int size)
{
    int p = size-1; int trans;
    int left =-1,right = size-1;
    while(left <= right)
    {
        left++;while(b[left] <= b[p]) left++;
        right--;while(b[right]>=b[p]) right--;
        if(right>=left)
        {
            trans = b[right];
            b[right] = b[left];
            b[left] = trans;
        }
    }
    else {trans = b[p];b[p] = b[left]; b[left] = trans;}
    return (left);
}
```

Section C: Design and Coding**[32 marks]**

17. Write a function that receives as input a date expressed as **d** (day) **m** (month) **y** (year), and returns the day of the week. For instance, if the date is **29/05/2014**, the function must return **Thursday**. You will use the following formula:

$$m_1 = \begin{cases} m - 2 & \text{if } m \geq 3 \\ m + 10 & \text{if } m < 3 \end{cases} \quad a_1 = \begin{cases} a & \text{if } m \geq 3 \\ a - 1 & \text{if } m < 3 \end{cases}$$

and with n_s being the two first digits of a_1 and a_s the two last digits of a_1 , we can compute f as

$$f = j + a_s + \frac{a_s}{4} - 2n_s + \frac{n_s}{4} + \frac{26m_1 - 2}{10}$$

The day of the week will then be given by the modulo of f and $7[f\%7]$ (Sunday for 0, Monday for 1, Tuesday for 2, Wednesday for 3, Thursday for 4, Friday for 5, Saturday for 6).

[12]

18. Numeric addresses for computers on the wide area network Internet are composed of four parts separated by periods, of the form **xx.yy.zz.mm**, where **xx**, **yy**, **zz**, and **mm** are positive integers. Locally computers are usually known by a nickname as well. You are designing a program to process a list of internet addresses, identifying all pairs of computers from the same locality (i.e, with matching **xx** and **yy** component).

- (a) Create a C structure called **InternetAddress** with fields for the four integers and a fifth component to store an associated nickname. [4]

- (b) Define a function, **ExtractInternetAddress**, that extracts a list of any number of addresses and nicknames from a data file whose name is provide as argument, and returns a dynamically allocated array that holds the indicated number of internet addresses (represented in **InternetAddress**) objects) retrieved from the file. The first line of the file should be the number of addresses that follow. Here is a sample data set:

```
4
111.22.3.44    plato
555.66.7.88    gauss
111.22.5.88    mars
234.45.44.88   ubuntu
```

[12]

- (c) Define a function **CommonLocality** that receives as arguments the array constructed in a) and the number of internet addresses, and displays a list of messages identifying each pair of computers from the same locality. In the messages, the computers should be identified by their nicknames. Here is a sample message:

```
Machines plato and mars are on the same local network.
```

[4]