

Eastern Mediterranean University

Department of Computer Engineering

**PRECEDENCE AND ASSOCIATIVITY**

**Instructions:**

* There are **5** questions in this **7** page sheet.

Please check !!!!!!

* Calculators are not allowed.
* GSM phones should be turned off.
* A table of operators for precedence and associativity is attached.
* Passing any material including rubbers, pencils etc. to anybody else is strictly prohibited in the exam.
* If an answer box is given in a question, you must give your answer (and nothing else) in the box !!!

|  |  |
| --- | --- |
| Question | Grade |
| P1(12 pnts.) |  |
| P2(12 pnts.) |  |
| P3(28 pnts.) |  |
| P4(24 pnts.) |  |
| P5(24 pnts.) |  |
| TOTAL: (out of 100) |  |

##### **Duration: 100 minutes**

*Name, Surname :…SOLUTION KEY………*

*Student No :………………………………*

*Group No :………………………………*

**Instructors:** *Prof.**Dr. Marifi Güler (Gr. 01, Gr. 02), Prof.**Dr. Omar Ramadan (Gr. 03, Gr. 04)*

CMPE 112 Final Exam

June 22, 2022

|  |  |
| --- | --- |
| Operators | Associativity |
| **( ) [ ] -> .** | **Left to right** |
| **! ++ -- + - \* & (type) sizeof** | **Right to left (Unary)** |
| **\* / %** | **Left to right** |
| **+ -** | **Left to right** |
| **< <= > >=** | **Left to right** |
| **== !=** | **Left to right** |
| **&&** | **Left to right** |
| **||** | **Left to right** |
| **?:** | **Right to left** |
| **= += -= \*= /= %=** | **Right to left** |
| **,** | **Left to right** |

***Some String Functions:***

***strlen(s1)*** Returns length of the string s1

***strcat( s1 , s2 )*** Concatenates a copy of string s2 onto the string s1

***strncat(s1, s2, n)*** Concatenates a copy of up to n characters from string s2 onto the string

in s1

**strcpy(s1, s2)** Copies the string s2 to s1

**strncpy(s1, s2, n)** Copies a string up to **n** characters from s2 to s1

***strcmp(s1,s2)*** Compares the string s2 with the string s1. Returns an integer less than, equal

to, or greater than depending on the result of the comparison

***strncmp(s1, s2, n)*** Compares at most the first n characters of the string s1 to s2, and

returns an integer less than, equal to , or greater than depending on the result of the

comparison

***strchr(s1 , c) L***ocates the first occurrence of c(a char) in the string s1, and returns a pointer

to the located character. If the search is not successful than null is returned

***strrchr(s1, c) L***ocates the last occurrence of c(a char) in the string s1, and returns a pointer

to the located character. If the search is not successful than null is returned

**Part I) [12 points] Multiple choice**

1. **Given the following declaration,**

void f(int x, int \*y, int \*z);

int x=3, y, z;

Which one of the following statements calls the above function correctly?

1. void f(int x, int &y, int &z);
2. void f(x, &y, &z);
3. f(int x, int &y, int &z);
4. f(x, &y, &z);
5. **Consider the following declaration:**

char s1[30] = "hi", s2[20] = "world", s3[10]= "";

The correct way to have string "hiworld" in s3 is

* 1. s3 = s1 + s2;
  2. strcpy(s3,s1); strcat(s3,s2);
  3. strcat(s3,s1); strcpy(s3,s2);
  4. strcpy(s3,s1); strcpy(s3,s2);

1. **What is the output of the following C code segment?**

if(strcmp("10","2")>0) printf("bigger");

else printf("smaller");

1. Bigger
2. Smaller
3. Not valid code fragment
4. None of the above answers
5. **Consider the following code fragment:**

char line[90];

int i;

gets(line);

Which one of the following is the most correct code fragment that will display just the characters typed by the user?

1. for(i=0;i<90;i++) printf("%c",line[i]);
2. for(i=0;i<89;i++) printf("%c",line[i]);
3. for(i=0;line[i]!= '\0';i++) printf("%c",line[i]);
4. for(i=0;i!='\0';i++) printf("%c",line[i]);

**Part II) [12points]** What is the output of the following code fragments:

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | char str1[12] = "Hello";  str1[2] = '\0';  printf("%d", strlen(str1)); | 2 |  |
| **2** | int A[3][3]={{1,2,3},{4,5,6},{7,8,9}};  int i,j,sum;  for(j=0; j< 3;j++) {  sum = 0;  for (i=0; i< 3;i++) {  if (i !=j)  sum = sum + A[i][j]; }  printf("%d ", sum);  } | 11 10 9 |  |
| **3** | int x[3]={2,5,9};  f(&x[2], x[0], &x[1]);  printf("%d %d %d\n", x[0],x[1],x[2]);    // The function f is defined as follows:  void f (int \*a, int b, int \*c) {  \*a = \*a + 2;  b = b - 1;  \*c = \*c + 1;} | 2 6 11 |  |
| **4** | for(i=1; i<=4; i++){  for(m=1; m<=5-i; m++)  printf("\*");  printf("\n");} | \*\*\*\*  \*\*\*  \*\*  \* |  |

**Part III) [28 points] Coding**

**1) [10 points]** Complete the following code that finds the maximum value in the last column of a 2-D array “x” and save it in variable max.

int x[3][4]={{2,20,-3,1},{4,-11,13,25},{15,42,6,-5}},max,i;

………… max=x[0][3];……………………………..

for(i=1; i<3; i++)

…………………… if(x[i][3]>max)…………………..

……… max = x[i][3];………………………………..

**2) [11points]** Write a function that receives two 1-D arrays of real values with **n** elements. The function computes (**not print**) their dot product.

Background: Given two vectors a = [a0, a1, a2, … , an-1] and b = [b0, b1, b2, … , bn-1], the dot-product is computed as

Let the function prototype be

void f(double a[], double b[], int n, double \*dp);

Complete the missing statements to achieve the required tasks.

**//function definition**

………void f1(double a[], double b[], int n, double \*dp)

{

double sum =0;

int i;

// compute the dot-product

for(i=0; i<=n;i++)

………………………………… sum+=a[i]\*b[i];

……………………………….. \*dp=sum

}

**3) [7points]** The following code reads a distance in miles from a file “data.txt” and convert it into kilometers, then the output will be written into a new file “result.txt”. Complete the missing statements to accomplish this task.

#include<stdio.h>

main() {

……………………………………………………………………………………………FILE \*inpfile, \*outfile;

double d\_miles, d\_kms;

inpfile=fopen("data.txt", "r");

if(………………………………………………………inpfile == NULL ………….){

printf("The file data.txt does not exist!");

exit(1);}

outfile=fopen("result.txt", "w");

while(…fscanf(inpfile, "%lf", &d\_miles) != EOF.…){

d\_kms=1.6\*d\_miles;

printf("%f mile is equivalent to %f kms\n", d\_miles, d\_kms);

fprintf(outfile,"%f mile is equivalent to %f kms\n",d\_miles, d\_kms);}

fclose(inpfile); fclose(outfile);

return 0;}

**Part IV) [24 points] Multiple choice.**

* + 1. Consider the function fun:

**float fun(float \*p, int n)**

**{**

**int k;**

**float a=0.0;**

**for(k=0;k<n;k++)**

**if(p[k] > 0.0) a+= p[k];**

**else a - = p[k];**

**return a;**

**}**

What does the function do when it is called as follows?

**float arr[100];**

**…**

**printf("%.2f\n", fun(arr+1, 100));**

1. It prints the sum of absolute values of the elements of **arr[100]**
2. It prints the sum of the positive elements of **arr[100]**
3. It prints the sum of absolute values of the elements of **arr[100]**, except the first element
4. It prints the sum of the positive elements of **arr[100]**, except the first element

----------------------------------------

**Ans: c**

* + 1. Consider the function fun:

**void fun(int \*pt, int n)**

**{**

**int \*q;  
 n--;**

**for(\_\_\_\_\_** MISSING CODE\_\_\_\_\_**) printf("%d ", \*q);**

**printf("\n");**

**}**

However, part of the code is missing (indicated by \_\_\_\_\_\_\_\_\_\_). The function is supposed to print the values of an array in reverse order (without changing the contents of the array). For example, when it is called as

**int array[5] = {1, 2, 3, 4, 5};**

**fun(array,sizeof(array)/sizeof(int));**

it will give the output **5 4 3 2 1**

What can the missing part be?

1. **q=pt+n; q >= pt; q**--
2. **\*q=\*pt+n; \*q >= \*pt; \*q**--
3. **q=pt+n; q >= pt; \*q**--
4. **\*q=\*pt+n; \*q >= \*pt; (\*q)**--

----------------------------------------

**Ans: a**

* + 1. Consider the code

**int k;**

**int \*p, \*q;**

**for(k=0; k<2; k++)**

**{**

**static int x = 1;**

**int y;**

**x++;**

**y = x +k;**

**p = &x;**

**q = &y;**

**}**

**\_\_**\_\_\_\_Dummy\_\_\_**\_\_\_\_\_\_\_**

**//** Code ends here

where the ‘Dummy’ is some statement. Which of the following can NOT be a substituted for the ‘Dummy’ (i.e. gives compiler error) ?

1. **printf("%d %d\n", \*p, \*q);**
2. **printf("%d\n", x);**
3. **(\*p)++**
4. **printf("%d\n", p-q);**

----------------------------------------

**Ans: b**

* + 1. Consider the declarations:

**float x[4][7];**

**float y[28];**

**char s[] = "ibm";**

**char q[] = {**'**i**'**,** '**b**'**,** '**m**'**};**

**int v[5] = {1,2};**

Which of the following is NOT correct?

1. Both **x** and **y** have **28** elements and occupy the same amount of memory
2. **v** has **5** elements
3. **s** has **3** elements
4. **q** has **3** elements

----------------------------------------

**Ans: c**

* + 1. Suppose that

**char \*x = String1, \*y=String2;**

Which expression will return **1** whenever String1 and String2 are the same strings?

a) **(x == NULL)**

b) **(x == y)**

c) **(x != y)**

d) **strcmp(x, y)**----------------------------------------

Ans: **c**

* + 1. Given the code

**FILE \*fp;**

**fp = fopen("abc.txt", "r");**

If fp is NULL which of the following may be true ?

a) Unable to open a file named abc.txt

b) abc.txt is not available on disk

c) Hard disk has hardware problems.

d) All of the above

----------------------------------------

**Ans: d**

**Part V) [24 points] Give the output of each question in the provided answer box.**

1. **void fun(int \*p, int x)**

**{**

**a = 6 b = 8**

**\*p += 4; x += 4;**

**}**

**main()**

**{**

**int a = 2, b = 8;**

**fun(&a, b);**

**printf("a = %d b = %d\n", a, b);**

**}**

----------------------------------------

1. **main()**

**{**

**int k, \*p, v[] = {2, 5, 3, 4, 6};**

**20 10 7 19 9**

**p = v + 2; \*p = 7;**

**p[2] = 9; p[-1] = 10;**

**v[0] = p[1] + 16;**

**\*(v+3) = \*p + 12;**

**for(k = 0; k <= 4; k++) printf("%d ", v[k]);**

**}**

----------------------------------------

**3) main()**

**{**

**C**

**CDEF**

**char p[] = "ABCDEF";**

**printf("%c\n", \*(p + \*p - 'B' + 3));**

**printf("%s\n", p + 2);**

**}**

----------------------------------------

**4)main()**

**{**

**void fun(void);**

**int k;**

**for(k=0; k < 2; k++)**

**fun();**

**}**

**4 4  
4 5**

**void fun(void)**

**{**

**int a = 3;**

**static int b = 3;**

**a++;**

**b++;**

**printf("%d %d\n", a, b);**

**}**

----------------------------------------

1. **main()**

**{**

**char a[] = "COMP\_ENG";**

**char \*v[2];**

**MP\_ENG**

**P**

**v[0] = a + 1;**

**v[1] = a + 2;**

**printf("%s\n", v[1]);**

**printf("%c\n", \*(v[0]+2));**

**}**

----------------------------------------

1. **main()**

**6 4 2**

**{**

**int \*ptr;**

**int array[] = {1, 2, 3, 4, 5, 6};**

**for(ptr=array+5; ptr >= array; ptr -= 2)**

**printf("%d ", \*ptr);**

**}**

== THE END ==

== YOU CAN USE THE SPACE BELOW FREELY ==