

## CMPE112/CMSE112 Questions with solutions Part II

**Q1)**

What is printed by this program? Answer in the box:

```
#include <stdio.h>

double g(double u)
{
    return u < 0 ? u - 1 : u + 1;
}

void f(double *px)
{
    *px += g(*px);
}

void main()
{
    double y = 5.2;
    f(&y);
    printf("\n %4.1f" , y);
}
```

11.4

**Q2)**

What is printed by this program? Answer in the box:

```
#include <stdio.h>
main()
{
    int *ptr;
    int array[4] = {1, 2, 3, 4};
    for(ptr=array+3; ptr >= array; ptr--)
        printf("%d ", *ptr);
}
```

4 3 2 1

**Q3)**

What is printed by this program? Answer in the box:

```
#include <stdio.h>
main()
{
    char *p = "ABC";
    printf("%c\n", *(p + *p - 'B' + 2));
}
```

B

**Q4**

What is printed by this program? *Answer in the box:*

```
#include <stdio.h>
void fun(int *p)
{
    int i;
    i = 1;
    while( i < 10)
    {
        *p = i;
        i *= 3;
        p++;
    }
}

main()
{
    int a[] = {8, 2, 9, 4, 6, 5, 7};
    int i;
    fun(a+2);
    for(i=0; i < 7; i += 2)
        printf("%d ", a[i]);
}
```

8 1 9 7

**Q5**

Fill in the blanks (denoted by \_\_\_\_ ) in the following program:

```
/* This program reads a value for x and, then, calls */
/* the function SetValue(). */
/* The function assigns the value 0 to x if x's entered value */
/* is less than 10; otherwise, it assigns the value */
/* 1 to x. */
```

```
#include <stdio.h>
main()
{
    void SetValue( _int *__ );
    int x;
    scanf("%d", &x);
    SetValue( _&x__ );
    printf("The new value of x is: %d", x);
}

void SetValue( _int *__ a)
{
    if ( __*a__ < 10) __*a__ = 0;
    else __*a__ = 1;
}
```

### Q6

Write a program that reads name and surname of a person from the keyboard. Then, if the name is lexicographically greater than the surname, it prints the name first and then the surname on the monitor. Otherwise, it prints the surname first and then the name. Note that name and surname cannot be more than 30 characters each and they are assumed to be typed in lowercase characters.

```
#include <stdio.h>
#include <string.h>
int main(){
char name[30], surname[30];
printf("Enter Name and Surname \n");
scanf("%s%s",name,surname);
if (strcmp(name,surname)>0) printf("%s %s", name , surname);
else printf("%s %s ",surname, name);
return 0;}
```

### Q7

(a) Fill in the blanks in the program given below which is used for sorting three numbers in ascending (increasing) order. For example, when prompted if the user enters

7.5 9.6 5.5

the program should print

The numbers in ascending order are: 5.50 7.50 9.60

Note: Lines that contain blanks are indicated by a > in the left margin.

```
/*
 * A program for ordering three numbers
 */
#include <stdio.h>
1> void order( _double*__smp, _double*__lgp);
int main(void) {
double num1, num2, num3; /* three numbers to put in order */
/* Gets test data */
printf("Enter three numbers separated by blanks> ");
scanf("%lf%lf%lf", &num1, &num2, &num3);
/* Orders the three numbers */
2> order(__&num1__, __&num2__);
3> order(__&num1__, __&num3__);
4> order(__&num2__, __&num3__);
/* Displays results */
printf("The numbers in ascending order are: %.2f %.2f %.2f\n",
num1, num2, num3);
return 0;
}
/*
 * Arranges arguments in ascending order
 */
5> void order(_double*__smp, _double*__lgp) {
double temp;
/* Compares values and swaps if necessary */
6> if (__*smp__ > __*lgp__) {
7> __temp__ = __*smp__;
8> __*smp__ = __*lgp__;
9> __*lgp__ = __temp__;
}
}
```

(b) What is the output of the following program?

```
#include <stdio.h>

void double_trouble(int *p, int y);
void trouble(int *x, int *y);

int main(void) {
    int x, y;
    trouble(&x, &y);
    printf("x = %d, y = %d\n", x, y);
    return 0;
}

void double_trouble(int *p, int y) {
    int x;
    x = 14;
    *p = 2 * x - y;
}

void trouble(int *x, int *y) {
    double_trouble(x, 5);
    double_trouble(y, *x);
}
```

Output

x=23, y=5

### Q8

Consider the following C programs. What is printed? Provide the exact form of the output as it is specified by the `printf()` statement.

```
#include <stdio.h>
void foo(int *);
main()
{
    int a [3]= { 1,2,3} ;
    foo(a);

    printf("%d\n", a[0]);
    printf("%d\n", a[1]);
    printf("%d\n", a[2]);
}

void foo( int *b)
{
    ++ b;
    *b =9;

    ++ b;
    *b *=2;
}
```

Output

1  
9  
6

### Q9

What is the output of the following program?

```

#include<stdio.h>
#include<string.h>
#define MAX 20
int main()
{
    char s1[MAX]="If you pass CMPE112";
    char s2[MAX]="we will meet";
    char *s3="next semester";
    char *s4="in CMPE212 course";
    char s5[3]="";
    printf("%d\n",strlen(s1));
    printf("%s\n",strchr(s2,'w'));
    printf("%s\n",strncat(s5,s4+8,2));
    printf("%s\n",strncpy(s5,s4,2));
    printf("%d\n",strncmp(s4+8,s1+17,2));
    //printf("%s-%s-%s-%s-%s\n",s1,s2,s3,s4,s5);
    return 0;
}

```

19  
will meet  
12  
in  
0

### Q10)

Write a C program for the following string of operation.

Read a string of data from the monitor and after calling **split** function , divide the given string into two parts as **first** and **last** and returns back to the main program.

**Split** function searches ‘\*’ character in the given string and copies all the characters before ‘\*’ character into **first** and copies all the remaining characters after ‘\*’ into **last** .

Afterwards compare **first** and **last** alphabetically in the main program and display the result as follow.

First is greater than last

First is less than last

First is equal to last

## Example

<b>book*abacus</b>	→input data	
alphabetically	<b>book</b> is greater than <b>abacus</b>	→ output of your program
abdullah*adem	→input data	
alphabetically	<b>abdullah</b> is less than <b>adem</b>	→ output of your program
<b>deniz*deniz</b>	→input data	
alphabetically	<b>deniz</b> is equal to <b>deniz</b>	→ output of your program

If necessary you can use some of the below string functions

**strlen(s1)**: Computes the length of the string s1, and returns the number of characters that precede ‘\0’.

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

**strcpy(s1, s2)**: Copies a string 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.

```
void split(char line[], char *first, char *last)
```

```
{int i, j, m;
```

```
for(i=0;line[i]!='*';i++)
```

```
    first[i]=line[i];
```

```
first[i]='\0';
```

```
for(j=i+1,m=0;j<=strlen(line);j++,m++)
```

```
    last[m]=line[j];
```

```
last[m]='\0';
```

```
}
```

```
int main()
```

```
{char t[30]; /* total string */
```

```
char f[20],l[20]; /* fist and last */
```

```

gets(t);
split(t , f , l);
if (strcmp(f,l)>0)
    printf("\n %s Greater than %s",f,l);
else if (strcmp(f,l)<0)
    printf("\n %s Less Than %s",f,l);
else printf("%s Equal to %s ",f,l);
return 0;
}

```

Q11)  
Trace and list the output of the program  
#include<stdio.h>

```

void fsp(int *x,int y)
{
    if (*x>y) {*x+=y;
               y+=*x;
              }
    else { y-=*x;
           *x=*x*y;
         }
    printf("X Value=%d and Y=%d\n",*x,y);
}

```

```

int main()
{
    int a=3,b=5;
    fsp(&a,b);
    a+=b;
    b+=a;
    printf("Result of A=%d and B=%d",a,b);
    return 0;
}

```

TRACE

a	b	*x	y	*x>y
3	5	3	5	false
			5-3=2	
			2	
		3*2=6		
		6		
6				
6+5=11				
	5+11=16			

OUTPUT

X Value =6 and Y=2  
Result of A=11 and B=16

Q12

Trace and list the output of the program

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

```
void fp(int a , int *y , int m)
{int i;
 for(i=0;i<m/2;i++)
 {
  if (i==0) *y=a+1;
  else *(y+i)--a;
  printf("I=%d and Y + %d = %d\n", i , i ,*(y+i));
 }
}
```

```
int main()
{
int array[3]={5,2,6};
int i,n=6;
fp(array[0],array,n);

for(i=0;i<n/2;i++)
printf(" %d \n",array[i]);
return 0;
}
```

#### TRACE

	array	n	a	m	i
y→	5--6	6	5	6	0
	2--4		4		1
	6--3		3		2
					3

#### OUTPUT

```
I=0 and Y+0=6
I=1 and Y+1=4
I=2 and Y+2=3
6
4
3
```

Q13)

The following function decides whether the (nxn) matrix is upper triangular or not by returning **1 if it is upper triangular and zero otherwise.**

(Note: a matrix is upper triangular if all elements below the diagonal are zero)

Example:

diagonal elements

4	5	3	4
0	7	0	6
0	0	5	3
0	0	0	7

Complete the missing part of the given function

```

int IsUpperTriangular(int matrix[m][m], int n)
{
    int i,j;
    for(i=1;i<n;i++){
        for(j=0;j<i;j++){
            if (matrix[i][j]!=0)
                return 0;
        }
    }
    return 1;
}

```

Q14

Using the following initializations in a program

```

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

```

What are the values of the following expressions ?

		<u>Values</u>
a)	*x ;	1
b)	*(x-1) ;	cancelled
c)	*x++ ;	1
d)	--*x ;	0
e)	x - y ;	0

Q15

What are the values of the following expressions ?

```

int arr[]={1 , 2 , 3 , 4 , 5};
int *parr[3];
parr[0]=arr;
parr[1]=arr+4;
parr[2]=parr[0]+2;

```

		<u>Values</u>
a)	*(parr[0]+1);	2
b)	*(parr[1]-1);	4
c)	*(parr[2]+1);	4

Q16

Given the following initializations:

```

int scores[] = {88, 98, 25, 53, 70, 66};
int *pscores = scores;

```

Write down the values of the following expressions:

- a) \*pscores .....88.....
- b) \*(pscores+2) .....25.....
- c) pscores – scores .....0.....
- d) (pscores[2] + 5) .....30.....
- e) --\*pscores .....87.....



### Q17

Given the following initializations

```
char word1[13] = "introduction";
char word2[14] = "toprogramming";
char str1[9] = "computer";
char str2[12] = "engineering"
```

What will be the values of the following expressions ?

- a) `printf("%d", strlen(word2));` .....13.....
- b) `printf("%s", strcat(word1, word2, 2));` .....introductionto.....
- c) `printf("%s", strncpy(word2, "algorithms", 9));` .....algorithm.....
- d) `printf("%d", strcmp(str1, str2));` .....-1.....
- e) `printf("%s", strcpy(str1, str2));` .....engineer.....

### Q18

Write a C code to do the following task: Read a series of characters from the standard input and write them to the standard output with the characters reversed, i.e., if the input is *Ahmet* then the output will be *temhA*.

Note: Do not use string functions

```
#include<stdio.h>

int main(void)
{
    char str[100], str_r[100];
    int n,i,k;

    printf("enter the a string of data\n");
    gets(str);

    for(n=0,i=0;str[i]!='\0';i++)
        n++;

    for(k=n-1,i=0;i<n;i++,k--)
        str_r[i]=str[k];

    str_r[i]='\0';

    printf("\n%s",str_r);

    return 0;
}
```