Lebanese University
Faculty of Sciences 1

INFO216 E

Imperative Programming II

Partial 11.12
Mon. 07 May pm
Duration: 1 hour

Exercise I

1. Draw memory state for the following program:

```
#include <stdio.h>

int f(int*,int);

int main(void){
        int t[]={2,3,4};
        int z=f(t,3);
        return 0;
}
```

```
int f(int *t,int s){
        int tmp,z;
        if(s<=0) return 0;
        if(s==1) return *t;
        tmp=*t;
        *t=*(t+1);
        *(t+1)=tmp;
        return z=*t*f(t+1,s-1);
}
```

Exercise II

2. Write the function printR that prints on the screen a given string in reverse order. You are not allowed to use "string.h" library. The recursive version is preferred, but not obligatory. Example: printR("animal"); prints on the screen: lamina

3. Write the function printReverse that prints the strings in an array in reverse order. This function uses the function printR. Below is printReverseTest(), and the resulting output.

```
void printReverseTest(){
        char *t[]={"animal","desserts",NULL,"gateman","","\0","depots"};
        printReverse(t,7);
}
int main(void) { printReverseTest(); return 0; }
```

Output on the screen (␣ indicates space) :
stoped␣␣␣nametag␣␣stressed␣lamina␣

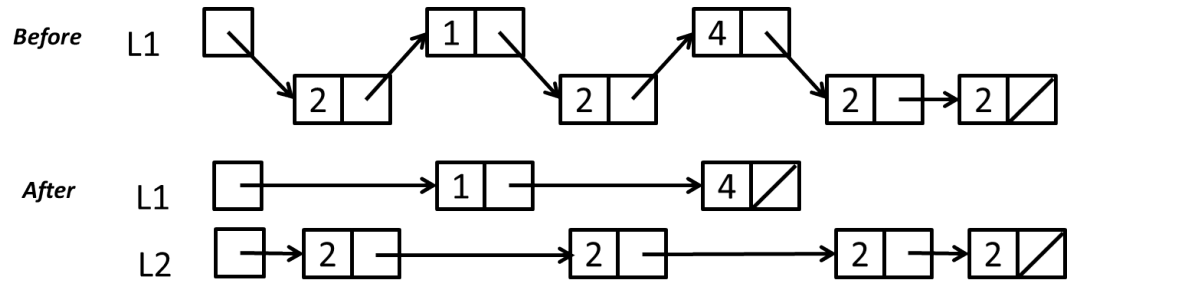
Exercise III

Consider the type node: typedef struct node{ int d; struct node* next;} node;

4. Write the function extract that extracts a sublist from a given list of integers. The nodes to be extracted are the node containing a given integer. No new allocation or deallocation is allowed.

Example:

```
node *L1,*L2=NULL;
//Push nodes to L1 to make the list {2,1,2,4,2,2}
extract(&L1,&L2,2);
```



The End