University of M'sila **Faculty** of mathematics and computer science **Department** of computer science

Level: 1st year of computer science
Course: ADS2

Exam

Academic year:2024/2025
Duration: 1h :30m

Exercise 1 : (8 pts)

a) Write a recursive function that takes a string `s` as input and returns the number of capital letter.

b) Write a C function that takes a string `s` (without spaces) as input and returns a new string with a space inserted before each capital letter.

Example: input s= "ThisIsATest", output a)4. Output b) t="This Is A Test".

tip: count the size of the new string before allocate it.

a) اكتب الدالة التراجعية التي تأخذ سلسلة رموز ثم تقوم بإرجاع عدد الحروف الكبيرة.

b) اكتب الدالة التي تأخذ سلسلة رموز بدون فراغات ثم تقوم بإرجاع سلسلة جديدة حيث تقوم بإضافة فراغ قبل كل حرف كبير

تلميح: قم بحساب حجم السلسلة الجديدة قبل حجزها.

Exercise 2 : (5 pts)

Write a C function that takes an odd integer n and returns a square matrix of size n x n. The matrix should contain sequential numbers along both diagonals from 1 to n . All non-diagonal elements should be 0.

Example: input n=5

output

1	0	0	0	1
0	2	0	2	0
0	0	3	0	0
0	4	0	4	0
5	0	0	0	5

اكتب دالة بلغة C تأخذ عددًا صحيحًا فرديًا n وتُرجع مصفوفة مربعة بحجم $n \times n$.

يجب أن تحتوي المصفوفة على أرقام متسلسلة من 1 إلى n على القطرين الرئيسيين (القطر الرئيسي والقطر الثانوي)، بينما تكون جميع العناصر غير الواقعة على الأقطار مساوية للصفر.

Exercise 3 : (7 pts)

Write a C function that takes a sorted singly linked list of integers and inserts all missing numbers between the smallest and largest values to make the sequence complete and continuous in ascending order.

Example:

Input: 1 → 4 → 5 → 7 → NULL

Output: 1 → 2 → 3 → 4 → 5 → 6 → 7 → NULL

اكتب دالة بلغة C تأخذ قائمة مرتبة أحادية من الأعداد الصحيحة، وتقوم بإضافة جميع الأعداد المفقودة بين أصغر وأكبر قيمة، بحيث تصبح القائمة متسلسلة بالكامل بترتيب تصاعدي.

Level: 1st year of computer science
Course: ADS2
Last name اللقب : ----------

Exam

First name الاسم : ----------

Academic year:2024/2025
Duration: 1h :30m
Groupe الفوج : ----------

Exercise 1 : (8 pts)

a) 2 pts

int countCap(**char*** s) {	
if(*s==’\0’) **return** 0;	
if (*s>=’A’ && *s<=’Z’)	
return 1+ countCap(s+1);	
return countCap(s+1);	
}	

b) 6 pts

char* insertSpaces(**char*** s) {	
char* r;	
int i, j, l, sp;	
l = strlen(s);	
sp = countCap(s);	
r = malloc(l + sp + 1);	
j = 0;	
for (i = 0; i < l; i++) {	
if (s[i]>=’A’ && s[i]<=’Z’ && i != 0)	
r[j++]=’ ’;	
r[j++] = s[i];	
}	
r[j] = '\0';	
return r;	
}	

Exercise 2 : (5 pts)

int** diagMat (**int** n) {	
int i, j, ** m;	
m = malloc(n * **sizeof**(int*));	
for (i = 0; i < n; i++){	
m[i] = malloc(n * sizeof int);	
for (j = 0; j < n; j++)	
m[i][j] =0;	
}	
for (i = 0; i < n; i++) {	
m [i][i] = i + 1;	
m [i][n - i - 1] = i + 1;	
}	
return m;	
}	

Exercise 3 : (7 pts)

void insertMissing(Node* h) {	
Node *t,*p;	
int i,diff;	
if(h==NULL||h->next==NULL) **return**;	
t = h;	
p=h->next;	
while (t && p) {	
diff = p->data - t->data;	
for(i=0;i<diff;i++) {	
add_head(t->next,t->data+1);	
t=t->next;	
}	
t=p;	
p = p->next;	
}	
}	