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:2023/2024
Duration: 1h :30m

Exercise 1 : (5 pts)

You have a 26-row amphitheater, with each row capable of accommodating either 3 or 4 students and the total number of students is a multiple of 5. Create a C function that calculates and displays the number of rows containing 3 students and the number of rows containing 4 students, assuming all rows are fully occupied. The function should consider all potential combinations of student distribution and should present the number of rows for each configuration along with the total number of students.

For example, the function will output:

nb of rows with 3 students = 4, nb of rows with 4 students = 22, and total number of students = 100.

nb of rows with 3 students = 9, nb of rows with 4 students = 17, and total number of students = 95.

لديك مدرجًا يتكون من 26 صفًا، حيث يمكن لكل صف استيعاب 3 أو 4 طلبة وان عدد الطلبة هو من مضاعفات العدد 5. قم بإنشاء دالة تحسب وتعرض عدد الصفوف التي تحتوي على 3 طلاب وعدد الصفوف التي تحتوي على 4 طلاب، بافتراض أن جميع الصفوف مشغولة بالكامل. يجب على الدالة أن تنظر إلى جميع التوزيعات المحتملة للطلاب ويجب أن تقدم عدد الصفوف لكل تكوين بالإضافة إلى إجمالي عدد الطلاب.

على سبيل المثال، الدالة ستظهر النتيجة بالشكل التالي:

عدد الصفوف التي تحتوي على 3 طلاب = 9، عدد الصفوف التي تحتوي على 4 طلاب = 17، وإجمالي عدد الطلاب = 95.

Exercise 2 : (3 pts)

write a recursive C function that takes a string ***s*** and an integer ***x*** as a key. This function encrypts the string by incrementing each character's ASCII value by the key. For instance, if the input string is "hello" and the key is 2, the encrypted string would be "jgnnq". Note that: 'h'+2='j', 'e'+2='g', ….. , 'o'+2='q'

اكتب الدالة التراجعية التي تأخذ جملة 's' والعدد x كمفتاح. تقوم هذه الدالة بتشفير السلسلة عن طريق إضافة العدد x لكل حرف من حروف الجملة. على سبيل المثال ، إذا كانت سلسلة الإدخال "hello" والمفتاح 2 ، فإن السلسلة المشفرة ستكون "jgnnq". علما ان 'h'+2='j' و 'e'+2='g' و..... و 'o'+2='q'

Exercise 3 : (7 pts)

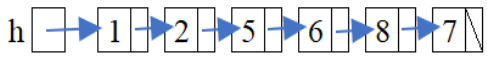
Write a C function that takes an array ***A*** and a number ***x***. This function should return an array that ends with the number **0** and excludes numbers that are multiples of ***x*** or divide ***x***.

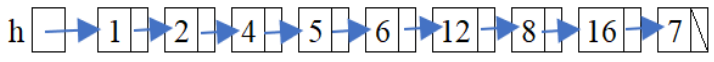
Example: A= {1, 2, 3, 4, 5, 6, 7, 8, 9} x= 4 result={3, 5, 6, 7, 9, 0}

اكتب دالة تأخذ جدولا ***A*** ورقما ***x***. يجب أن ترجع هذه الدالة جدولًا ينتهي بالرقم 0 ويستثني الأرقام التي هي من مضاعفات ***x*** أو تقسم ***x***.

Exercise 4 : (5 pts)

Write a C++ function that takes a list of integers and modifies the list by adding a new item after each element that is even. The new item should be the double of the even number. You may utilize the following pre-defined functions from the course without needing to redefine them: 'add_head', 'append_end', 'delete_head', and 'delete_end'.

Example: 



اكتب دالة تاخذ قائمة من الأعداد الصحيحة وتعدل القائمة بإضافة عنصر جديد بعد كل عنصر زوجي. العنصر الجديد يجب أن يكون ضعف العدد الزوجي.

يمكنك استخدام الدوال المعرفة مسبقاً في الدرس دون الحاجة إلى إعادة تعريفها: "add_head'، 'append_end'، "delete_head'، و 'delete_end'.

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

Exam

First name الاسم : ----------

Academic year:2023/2024
Duration: 1h :30m
Groupe الفوج : ----------

Exercise 1 : (5 pts)

void calculateRows () {	
int nb3, nb4, total;	
for (nb3=0;nb3<=26;nb3++){	
nb4=26-nb3;	
total=nb3*3+nb4*4;	
if (total%5==0)	
printf("nb of rows with 3 students = %d, nb of rows with 4 students = %d, and total number of students = %d\n",nb3,nb4,total);	
}	
}	

Exercise 2 : (3 pts)

void encryptStr(**char** *s, **int** x){	
if(*s=='\0') **return**;	
*s=*s+x;	
encryptStr(s+1, x);	
}	

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

Exam

First name الاسم : ----------

Academic year:2023/2024
Duration: 1h :30m
Groupe الفوج : ----------

Exercise 3 : (7 pts)

int* filteX (**int*** A, **int** n, **int** x){	
int* t, i, j;	
t=(**int***)malloc(n***sizeof**(**int**));	
j=0;	
for (i=0;i<n;i++)	
if(A[i]%x==0|| x % A[i] ==0){	
if(A[i]%x!=0 && x % A[i] !=0){	
j++;	
}	
t[j]=0;	
t=(**int***)realloc(t, (j+1)***sizeof**(**int**));	
return t;	
}	

Exercise 4 : (5 pts)

void insertEven(List h) {	**void** insertEven(List h) {
while(h!=NULL){	Node* e;
if(h->data%2==0){	**while**(h!=NULL){
add_head(h->next, h->data*2);	**if**(h->data%2==0){
h= h->next;	e=**new** Node ;
	e->data = h->data*2;
}	e->next=h- > next ;
h= h->next;	h->next=e;
}	h=e;
}	}
	h= h->next;
	}
	}