

Exercise 1 : (4.75 pts)

Write the "sum" function to calculate the following sum without using the pow function:

اكتب الدالة "sum" لحساب المجموع التالي دون استخدام دالة pow:

$$S = \sum_{i=0}^n \frac{x^{n-i}}{2n-i}$$

Exercise 2 : (3.25 pts)

Create a recursive function that takes a string 's' and gives back the first capital letter encountered. For example, if the input is 'hello World', the function should return 'W'. if there is no capital letter, it returns '\0'.

أنشئ دالة ترارجية تأخذ سلسلة رموز 's' وتعيد أول حرف كبير تجده في النص. على سبيل المثال، إذا كان المدخل 'hello World'، يجب أن تُرجع الدالة الحرف 'W'. إذا لم يكن هناك أي حرف كبير، فستُرجع '\0'.

Exercise 3 : (7.25 pts)

Write a function that takes two arrays, 't1' and 't2', each with 'n' elements. These arrays represent binary numbers. The function should return an array representing the bitwise sum of the two binary numbers without carrying over (ignoring the last carries).

اكتب دالة تأخذ جدولين، 't1' و 't2'، كل منهما يحتوي على 'n' عنصرًا. يُمثل هذان الجدولان أرقامًا ثنائية. يجب على الدالة أن تُرجع جدولًا يمثل مجموع بتات للرقمين الثنائيين دون مراعاة الفائض الأخير (باللون الرمادي).

For example:

carry	1		1		1
t1	1	0	1	1	0
+					
t2	1	1	0	1	0
=					
t3	1	0	0	0	0

Exercise 4 : (4.75 pts)

Write a C++ function that takes a list of integers and a number 'n'. This function should remove 'n' elements from the beginning of the list and add them to the end. 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'.

اكتب دالة C++ تأخذ قائمة من الأعداد الصحيحة والرقم 'n'. يجب على هذه الدالة إزالة 'n' عناصر من بداية القائمة وإضافتها إلى النهاية. يمكنك استخدام الدوال المعرفة مسبقًا في الدورة التالية دون الحاجة إلى إعادة تعريفها: 'add_head', 'append_end', 'delete_head', و 'delete_end'.

Example:

Original list: 'h = 10 -> 20 -> 30 -> 40 -> 50 -> NULL'

n = 2

Resulting list: 'h = 30 -> 40 -> 50 -> 10 -> 20 -> NULL'

Good luck

Exercise 1 : (4.75 pts)

float sum(float x, int n) {	
float s, b;	
int i;	
s=0;	
b=1;	
for (i=n; i>=0; i--){	
s=s+b/(2*n-i);	
b=b*x;	
}	
return s;	
}	

Exercise 2 : (3.25 pts)

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

Exercise 3 : (7.25 pts)

int* sumBinnary (int* t1, int* t2, int n){	
int i, r, x, *t3;	
t3=(int*)malloc(n*sizeof(int)) ;	
r=0;	
for(i=n-1; i>=0;i--){	
x= t1[i]+ t2[i]+r;	
t3[i]= x%2;	
r=x/2;	
}	
return t3;	

}	
---	--

Exercise 4 : (4.75 pts)

void rotateList(List &h, int n) {	
int x, i;	
if(h!=NULL)	
for(i=0 ; i<n ;i++){	
x=h->data;	
delete_head(h);	
append_end(h, x);	
}	
}	