情報処理特論I 第6回課題

5J04 岩埼悠紀

＜演習課題2＞

1. “You are given an array of integers (both positive and negative). Find the continuous sequence with the largest sum. Return the sum.”（第4回実施）について、以下の演習を行ってください。
2. 授業時に考えたアルゴリズム（3.の解答）を実装し、正しく出力されることを確認しなさい。（標準）出力には、入力数列【INPUT: {x1, x2, …}】と出力値【OUTPUT: Xa】が必ず表示し、wordファイル上に添付してください。（単一の数値を和に定義しても良い）

（＋α）

余裕があれば、出力上に当該の配列も出力できるようにしてください【OUTPUT: Xa ({, , …})】。

＜結果＞

|  |
| --- |
| #include <stdio.h>  int global\_start=0;  int global\_end=1;  int getMaxSum(int a[], int size) {  int i;  int max\_sum = a[0];  int sum = a[0];  for (i = 1; i < size; i++) {  sum += a[i];  if (max\_sum < sum){  max\_sum = sum;  gloal\_end++;  }  if(sum < a[i]){  sum = a[i];  global\_start = i;  }  printf("max\_sum = %d\n",max\_sum);  }  return maxSum;  }  int main(void) {  int i;  int size = 5;  int d[] = {1,2,-4,5,3};  printf("INPUT:{");  for (i = 0; i < size-1; i++) {  printf("%d,", d[i]);  }  printf("%d",d[i]);  printf("}\n");  printf("OUTPUT:%d {",getMaxSum(d, size));  for (i = global\_start; i < global\_end; i++) {  printf("%d,", d[i]);  }  printf("%d",d[i]);  printf("}\n");  } |



図１　課題1-Aの出力

1. “sum”→”product”に変更し、授業で考えたアルゴリズム（4.の解答）を実装してください。

＜結果＞

|  |
| --- |
| #include <stdio.h>  int global\_start=0;  int global\_end=0;  int getMaxProduct(int a[], int size) {  int i;  int max\_product=0, product=1, length=0;  for(i=0;i<size;i++){  if(a[i] == 0){  product = 1;  length=0;  }else{  product \*= a[i];  if(max\_product < product){  max\_product = product;  global\_end = i;  length++;  }else if(max\_product < a[i]){  max\_product = a[i];  }  }  }  global\_start = global\_end-length-1;  return max\_product;  }  int main(void) {  int i, size = 6;  int data[] = {1,-1,5,5,-2,-1};  printf("INPUT:{");  for (i = 0; i < size-1; i++) {  printf("%d,", data[i]);  }  printf("%d",data[i]);  printf("}\n");  printf("OUTPUT:%d {",getMaxProduct(data,size));  for (i = global\_start; i < global\_end; i++) {  printf("%d,", data[i]);  }  printf("%d",data[i]);  printf("}\n");  } |



図２　課題1-Bの出力

1. 前問 (1-B.)のアルゴリズムは、実際には不完全なアルゴリズムです。なぜ、不完全であるかを具体的な入力数列と出力値の例を挙げて、その場合の標準出力をwordファイル上に添付してください。

前から順番に処理を行っているため，最後のマイナス以降を除いたものと，最初のマイナス以前のものの比較をしていないので，正しい結果が得られていない。

1. では、どのように改良すれば例外なく正しい出力となるアルゴリズムになるか考え、実装してください。

|  |
| --- |
| #include <stdio.h>  int global\_start=0;  int global\_end=0;  int getMaxProduct(int a[], int size) {  int i;  int max\_product = 0;  int product = 1;  int l = 0;  int h = 1;  for(i=0;i<size;i++){  if(a[i] == 0){  product = 1;  l=0;  }else{  product \*= a[i];  if(h>0 && product <0){  h = product;  global\_start = i+1;  }  if(max\_product < product){  max\_product = product;  global\_end = i;  l++;  }else if(max\_product < a[i]){  max\_product = a[i];  }  }  }  if(max\_product < (product/h)){  global\_end = size-1;  max\_product = product/h;  }else{  global\_start = global\_end-l-1;  }  return max\_product;  }  int main(void) {  int i;  int size = 6;  int data[] = {1,-3,3,1,-2,8};  printf("INPUT:{");  for (i = 0; i < size-1; i++) {  printf("%d,", data[i]);  }  printf("%d",data[i]);  printf("}\n");  printf("OUTPUT:%d {",getMaxProduct(data,size));  for (i = global\_start; i < global\_end; i++) {  printf("%d,", data[i]);  }  printf("%d",data[i]);  printf("}\n");  } |



図３　課題1-Dの出力

1. “Write a method to generate a random number between 0 and 6, given a method that generates a random number between 0 and 4. (i.e. implement rand7() using rand5()).”（第5回実施）について、以下の演習を行ってください。
2. 授業で取り扱った3つのアルゴリズム（2. & 3. & 4.）をそれぞれ実装してください。また、呼び出し回数非決定法（3. & 4.）のアルゴリズム2つについては、rand5()関数の呼び出し回数を測定して、それぞれの平均呼び出し回数を求めてください。

|  |
| --- |
| #include<stdio.h> #include "MT.h"  int global\_counter3 = 0; int global\_counter4 = 0;  int rand5() {  return genrand\_int32() % 5; }  int rand7\_2(){  int v = rand5() + rand5();  return (v%7+1); }  int rand7\_3(){  while(1){  int num = 5 \* rand5() + rand5();  global\_counter3 = global\_counter3 + 2;  if(num < 21) return (num%7)+1;  } }  int rand7\_4(){  while(1){  int num = 2\*rand5();  int odd = rand5();  global\_counter4 = global\_counter4 + 2;  if(odd != 4){  num = num + odd %2;  if(num < 7) return num+1;  }  } }  int main(void) {  printf("乱数生成結果\n");  printf("2 { ");  for (int i = 0; i < 10; i++) {  printf("%d ",rand7\_2());  }  printf("}\n");   printf("3 { ");  for (int i = 0; i < 10; i++) {  printf("%d ",rand7\_3());  }  printf("}\n");   printf("4 { ");  for (int i = 0; i < 10; i++) {  printf("%d ",rand7\_4());  }  printf("}\n");   printf("rand5()呼び出し回数平均（試行回数1000）\n");  global\_counter3 = 0;  global\_counter4 = 0;   for(int i=0;i<1000;i++){  rand7\_3();  rand7\_4();  }  printf("3 :%lf\n",(double)global\_counter3/1000);  printf("4 :%lf\n",(double)global\_counter4/1000);  return 0; } |

[結果]

2{5 5 5 7 1 1 2 5 1 4}

3{1 5 6 7 1 3 2 5 1 1}

4{2 5 3 1 7 5 3 2 2 7}

Rand5()呼び出し回数平均（試行回数）1000

3:2.380000

4:3.694000

1. 次のアルゴリズムを実装してください。

“Write a method to generate a random number between 0 and 14, given a method that generates a random number between 0 and 6. (i.e. implement rand15() using rand7()).”

|  |
| --- |
| #include<stdio.h>  #include "MT.h"  int rand5() {  return genrand\_int32() % 5;  }  int rand7(){  while(1){  int num = 5 \* rand5() + rand5();  if(num < 21) return (num%7)+1;  }  }  int rand15(){  while(1){  int num = 7 \* rand7() + rand7();  if(num < 45) return (num%15)+1;  }  }  int main(void) {  int I;  for(i=0;i<20;i++){  printf("%d ",rand15());  }  return 0;  } |