学位プログラム	プログラム							
受験番号	解	答	例					

合計点	

チェック欄

※この試験科目を解答する場合

チェック欄に✔をつけてください。

令和８年度

新潟大学工学部第３年次編入学（第２次募集）

学　　力　　試　　験

試 験 科 目	専門基礎科目（プログラミング）	全２頁 （表紙を除く）

注意事項

１．表紙の所定欄に志望する学位プログラム名，受験番号を記入してください。

２．解答はその問題と同一の試験用紙に記入してください。解答スペースが足りない場合は，「(裏面に続く)」と明記したうえで，その用紙の裏に続けて解答してください。また，選択しなかった科目は，表紙にのみ受験番号を記入してください。

３．試験用紙の所定欄に受験番号を必ず記入してください。

４．各プログラムで解答する科目は以下の表の通りです。科目の選択があるプログラムは表をよく確認の上，科目の過不足がないように注意してください。

５．選択した答案には表紙の左上のチェック欄に✔を付けてください。✔がない答案は採点されません。

６．本冊子のホチキス止めは外さないでください。

学位プログラム	学力試験科目（専門基礎科目）
社会基盤工学プログラム	「数学，物理」の２科目
知能情報システムプログラム	「数学，プログラミング」の２科目
化学システム工学プログラム （化学工学コース）	「化学（〔Ⅱ〕無機化学，〔Ⅲ〕物理化学，〔Ⅳ〕化学工学）」
材料科学プログラム	「化学（〔Ⅰ〕有機化学，〔Ⅱ〕無機化学，〔Ⅲ〕物理化学）」 もしくは「数学，物理」の２科目
建築学プログラム	「数学，物理」の２科目

受験番号	解答例

令和８年度

新潟大学工学部第３年次編入学（第２次募集）

学　　力　　試　　験

試 験 科 目	専門基礎科目（プログラミング）	１／２頁

〔Ｉ〕以下の問いに答えよ。解答には，プログラムの内容や考え方についての説明を書き加えてもよい。

異なる自然数 n,m に対して，それぞれの自身を除いた正の約数の和が互いに他方と等しいとき，自然数 n,m の組を友愛数の組と呼ぶ。10000 までの数の中での友愛数の組を全て出力するプログラムを C 言語で書け。

```
#include <stdio.h>
int sum_of_divisors(int n) {
  int i, sum = 0;
    for (i = 1; i < n; i++) {
        if (n % i == 0) {
            sum += i;
        }
    }
    return sum;
}

int main(void) {
  int i, a, b;
  int divisor_sums[10000];

  for (i = 1; i <= 10000; i++) {
    divisor_sums[i-1] = sum_of_divisors(i);
  }

  for (a = 1; a <= 10000; a++) {
    b = divisor_sums[a-1];
    if (b <= 10000) {

     if (a < b && divisor_sums[b-1] == a) {
        printf("(%d, %d)¥n", a, b);
     }
    }
  }
  return 0;
}
```

受験番号	解答例

令和８年度
新潟大学工学部第３年次編入学（第２次募集）
学　　力　　試　　験

試 験 科 目	専門基礎科目（プログラミング）	２／２頁

〔Ⅱ〕以下の問いに答えよ。解答には，プログラムの内容や考え方についての説明を書き加えてよい。

図１に人工ニューロンの構造を示す。人工ニューロンは n 個の入力値 x_i $(i = 1, 2, \ldots, n)$ を受け取り，それぞれの入力に対して対応する重み ω_i を掛けた値の総和にバイアス項 b を加え，さらにその結果にシグモイド関数 $f(x) = \frac{1}{1+e^{-x}}$ を適用することで出力値 y を求める。実数の計算は float 型を用いて行う。

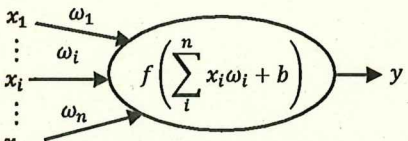

図１．人工ニューロンの構造

（１）　任意の入力数の人工ニューロンを表現する構造体 struct Neuron を C 言語で定義せよ。この構造体には，入力の数，重みを格納する任意長の配列，およびバイアス項を保持するためのメンバーを含めよ。

（２）　n 個の入力値を格納した配列 inputs に対する入力数 n の人工ニューロンの出力値を計算する関数
float Compute_Output(struct Neuron* neuron, float* inputs)
を C 言語で書け。e^x を計算する関数 float expf(float x) を用いてよい。

（１）

```
struct Neuron {
    int n;
    float* weights;
    float bias;
};
```

（２）

```
float Compute_Output (struct Neuron* neuron, float* inputs) {
    int i;
    float sum = neuron->bias;
    for (i = 0; i < neuron->n; i++)
        sum += inputs[i] * neuron->weights[i];
    return 1.0 / (1.0 + expf(-sum));
}
```