

Zaawansowane języki programowania
Laboratorium 12 (Adresowanie oraz implementacja komend strukturalnych)

Zadanie 1. Jakie są rozmiary poniższych struktur?

```
typedef struct{  
    int a;  
    char b;  
    char c;  
} QQ4;
```

```
typedef struct{  
    char a;  
    int b;  
    char c;  
} QQ5;
```

Zadanie 2. Załóżmy, że w programie zadeklarowano następującą strukturę:

```
typedef struct{  
    int a;  
    double b;  
} QQ;
```

Niech będzie zadeklarowana tablica:

```
QQ tab[20][30];
```

Załóżmy, że rozpoczyna się ona od adresu 5000. Które bajty pamięci zajmą poniższe zmienne?

- | | |
|---------------|-----------------|
| a) tab[0][0] | c) tab[0][0].a |
| b) tab[5][10] | d) tab[5][10].b |

Zadanie 3. Załóżmy, że w programie zadeklarowano następującą strukturę:

```
#pragma pack(push)  
#pragma pack(1)  
typedef struct{  
    int a;  
    double b;  
} QQ;  
#pragma pack(pop)
```

Niech będzie zadeklarowana tablica:

```
QQ tab[10][15];
```

Załóżmy, że rozpoczyna się ona od adresu 2000. Które bajty pamięci zajmą poniższe zmienne?

- | | |
|---------------|-----------------|
| a) tab[0][0] | c) tab[0][0].a |
| b) tab[5][10] | d) tab[5][10].b |

Zadanie 4. Załóżmy, że w programie zadeklarowano następującą strukturę:

```
#pragma pack(push)
#pragma pack(1)
typedef struct{
    int a:16;
    int b:4;
    int c:4;
} QQ;
#pragma pack(pop)
```

Niech będzie zadeklarowana tablica:

```
QQ tab[8][20];
```

Założmy, że rozpoczyna się ona od adresu 3000. Które bajty pamięci zajmą poniższe zmienne?

- | | |
|---------------|-----------------|
| a) tab[0][0] | c) tab[0][0].a |
| b) tab[5][10] | d) tab[5][10].b |

Zadanie 5. Załóżmy, że w programie zadeklarowano następującą strukturę:

```
typedef struct{
    double a;
    char b[10][15];
    int c[12];
} ABC;
```

Niech będzie zadeklarowana tablica:

```
ABC tab[10][100][20];
```

Założmy, że rozpoczyna się ona od adresu 1000. Które bajty pamięci zajmą poniższe zmienne?

- | | |
|-----------------|-------------------------|
| a) tab[0][0][0] | c) tab[2][3][4].b[2][3] |
| b) tab[2][3][4] | d) tab[2][3][4].c[10] |

Zadanie 6 (0.5 pkt). Napisać program, który będzie obliczał adresy struktur dla powyższych przykładów. Argumentami wejściowymi są początkowy adres pamięci oraz indeksy tablicy. Reszta danych czyli struktury, rozmiary tablicy mogą być wpisane na sztywno w programie.

ZADANIE DOMOWE

Edgar Dijkstra zaproponował kiedyś komendę, która łączyła cechy pętli **while** i **switch**. Jej semantyka jest następująca:

```
do
    <Warunek1> -> <komenda>
    <Warunek2> -> <komenda>
    <Warunek3> -> <komenda>
    .
    .
    .
    <WarunekN> -> <komenda>
od
```

Jeżeli zachodzi warunek to wykonywana jest komenda. W przypadku kiedy zachodzi wiele warunków, wybierana jest niedeterministycznie jedna z komend. Pętla kończy działanie, kiedy nie zachodzi żaden z warunków.

Oprócz tego można spotkać również następującą konstrukcję:

```
if
  <Warunek1> -> <komenda>
  <Warunek2> -> <komenda>
  <Warunek3> -> <komenda>
  .
  .
  .
  <WarunekN> -> <komenda>
fi
```

W literaturze składnie te nazywa się **guarded commands**.

Wykorzystując tę strukturę możemy bardzo czytelnie napisać na przykład znalezienie minimum z dwóch liczb a, b :

$(a \geq b) \rightarrow \min = b$

$(a < b) \rightarrow \min = a$

Zadanie 7. Używając **guarded commands** napisz programy, które:

- a) (0.5 pkt) Niech dany będzie ciąg A składający się z czterech liczb a_1, a_2, a_3, a_4 . Zaimplementować program, tak aby w wyniku otrzymano posortowany ciąg A .
- b) (1 pkt) Niech dane będą liczby $a, b \in \mathbb{N}$. Zaimplementować program liczący $NWD(a, b)$.

TERMIN: Dwa dni przed kolejnym laboratorium.

Rozwiązania przesłać na adres: mmiotk@inf.ug.edu.pl.