

## Przykładowe tematy z JiMP

1. Proszę napisać program, który dokona konwersji swojego argumentu wywołania z punktów na centymetry, ( $77.27 \text{ pt} = 1 \text{ cal} = 2.54 \text{ cm}$ ) tzn. np. wywołanie:  
`c:\>pkt 144.54` spowoduje wypisanie:  
`= 5.08 cm`
2. Proszę napisać program, który dokona konwersji swojego argumentu wywołania, ze stopni Celsjusza na stopnie Fahrenheita ( $c = 5/9*(f-32)$ )  
`c:\>cel 10` spowoduje wypisanie:  
`= 50 st. F`
3. Proszę napisać program, który dokona konwersji swojego argumentu wywołania z centymetrów na stopy i cale ( $30\text{cm}=1\text{ft}$ ,  $2.54\text{cm}=1\text{in}$ ), tzn. na przykład wywołanie:  
`c:\>cm 182` spowoduje wypisanie:  
`około 6 stop i 1 cal`
4. Proszę napisać program, zamieniający wszystkie litery na małe (czyta z `stdin`, pisze na `stdout`).
5. Proszę napisać funkcję  
`int bc( unsigned int n );`  
zliczając zapalone (czyli  $= 1$ ) bity w `n`.
6. Proszę napisać program wypisujący na `stdout` pierwszych 15 linii z pliku, którego nazwa podana jest jako argument tzn. na przykład wywołanie:  
`c:\>p15 jakisplik.txt`  
spowoduje wypisanie na ekran pierwszych 15 linii z pliku `jakisplik.txt`.
7. Proszę napisać program, który zlicza znaki w podanym jako pierwszy argument słowie i wypisuje je na `stdout`, tzn. wywołany w postaci  
`c:\>ile abcdefgh` spowoduje wypisanie:  
`8`
8. Proszę napisać **funkcję**  
`double *max( int n, double v[] )`  
zwracającą wskaźnik do maksymalnego elementu wektora `v`. Jak zachowuje się Twoja funkcja, jeśli w wektorze jest kilka elementów o maksymalnej wartości ?
9. Proszę napisać program, który odwraca kolejność liter w podanym jako argument słowie i wypisuje je na `stdout`, tzn. wywołany w postaci  
`c:\>wspak abcdefgh` spowoduje wypisanie:  
`hgfedcba`
10. Proszę napisać funkcję, która skleja dowolną liczbę ciągów znaków w jeden:  
`char *napis= sklej( "Ala ", "ma ", "kota.", "" );`  
Proszę założyć, że wynik nie będzie dłuższy niż 128 znaków.

11. Do czego służy plik `stdarg.h`. Proszę wykorzystać ten mechanizm do napisania funkcji, która sumuje dowolną liczbę dodatnich liczb `double`:

```
double trzynascie= suma( 0.0, 5.5, 1.25, 2.25, 4., -1. );
```

Ostatnia liczba jest ujemna i służy tylko do zaznaczenia końca sumowanego ciągu.

12. Dana jest tablica `t` zawierająca struktury:

```
typedef struct {
    int key;
    string signature;
} Elem;
```

```
Elem t[1000];
```

Napisz funkcję porównującą do sortowania `t` rosnąco ze względu na pole `key` przy pomocy funkcji standardowej `qsort`.

13. Proszę napisać funkcję `nazwa`, która pobiera ciąg znaków i liczbę całkowitą i zwraca nazwę pliku zgodnie z przykładem:

```
char *fn= nazwa( "c:\temp\moj", 1 )
FILE *f= fopen( fn, "w" )
```

spowoduje otwarcie do pisania pliku o nazwie `c:\temp\moj.1`

14. Dana jest tablica `t` zawierająca struktury:

```
typedef struct {
    int key;
    float val;
} Elem;
```

```
Elem t[1000];
```

Napisz funkcję porównującą do wyszukiwania w `t` elementów o danej wartości pola `key` przy pomocy funkcji standardowej `bsearch`.

15. Proszę pokazać na przykładzie jak wygląda i do czego służy prototyp funkcji? Czy prototyp funkcji może wystąpić w tym samym pliku, co definicja funkcji? Jeżeli tak, to proszę zademonstrować przydatność takiej konstrukcji.

16. Proszę zdefiniować typy (napisać odp. deklar. `typedef`):

- „wskaźnik do funkcji pobierającej argument typu `int` i zwracającej obiekt `double`”,
- „wskaźnik do funkcji pobierającej wskaźnik na tablicę napisów i zwracającej tablicę liczb `int`”,
- „wskaźnik do struktury zawierającej: liczbę `int`, napis o „dowolnej” długości i wskaźnik do takiej samej struktury”
- „tablice `pi`©ciu wskaźników do funkcji pobierającej argument typu `int` i zwracającej obiekt `int`”.

17. Jaki ciąg znaków zostanie wypisany w wyniku działania następującego fragmentu programu:

```
int a= 12, b= 16, c= 2;
a -= b + c >> 2;
printf( " a is %d\n", a );
```

Odpowiedź proszę uzasadnić. Uwaga: odpowiedź nie jest trywialna.

18. Jaki ciąg znaków zostanie wypisany w wyniku działania następującego fragmentu programu (program daje się skompilować i jest poprawny):

```
#define MAX 3
main() {
    int i;
    int a[] = { 0,1,2,3,4,5};
#ifdef DEBUG
    for( i= 0; i < MAX; i++ )
        printf( " %d\n", a[i] );
#else
    printf( "Done!" );
#endif
    printf( "\n" );
    ...
}
```

Odpowiedź proszę uzasadnić.

19. W wyniku próby kompilacji programu:

```
#include <stdio.h>
main() {
    static int ar[]={ 1,2,3,4,5,6,7,8,9,0};
    int a= 5++;
    printf( "%d\n", ar[a] );
    return 0;
}
```

Pojawił się następujący komunikat o błędzie:  
Error xx.c 4: Lvalue required in function main  
Proszę go wyjaśnić.

20. Proszę przeanalizować fragment programu:

```
int *a, *b;
int x= 5, y= 10;
a= &x;
*b *= *a + x - y;
printf( "%d\n", *b );
```

Jakie mogą być skutki działania tego kodu oraz jaki ciąg znaków zostanie wypisany na stdout. Odpowiedź proszę uzasadnić.

21. Proszę napisać makroinstrukcję ZAM( i, j, v ) równoważną funkcji

```
void zamien( int i, int j, double v[] ) {
    double tmp= v[i];
    v[i]= v[j];
    v[j]= tmp;
}
```

22. Jaki ciąg znaków zostanie wypisany w wyniku działania następującego programu:

```
#include <stdio.h>
int count( int x ) {
    static int log;
    log += x;
    return log;
}
main() {
    int i;
    for( i= 1; i <= 30; ) {
        i= count( i );
        printf( " %d", i );
    }
    return 0;
}
```

Odpowiedź proszę uzasadnić.

23. Jaki ciąg znaków zostanie wypisany w wyniku działania następującego programu:

```
#include <stdio.h>
int val( int x ) {
    while( x < 5 )
        x++;
    return x << 3;
}
main() {
    int s, a= 1;
    s= val( a );
    printf( "s= %d\n", s );
    return 0;
}
```

Odpowiedź proszę uzasadnić.

24. Jaki ciąg znaków zostanie wypisany w wyniku działania następującego programu:

```
#include <stdio.h>
void sw( int x, int *y ) {
```

```

        int t= x;
        x= *y;
        *y= t;
    }
    main() {
        int a= 5, b= 10;
        sw( &a, b );
        printf( "a= %d  b= %d\n", a, b );
        return 0;
    }

```

Odpowiedź proszę uzasadnić.

25. Jaki ciąg znaków zostanie wypisany w wyniku działania następującego programu:

```

#include <stdio.h>
cnt( int *a, int n ) {
    int p= 0;
    while( --n >= 0 )
        p += a[n] > 0 ? 1 : 0;
    return p;
}
main() {
    int a[]= {1,2,3,-4,5,-6};
    int ne= (sizeof a / sizeof a[0]);
    int nd= cnt( a, ne );
    printf( "%d elements, %d positive\n", ne, nd );
    return 0;
}

```

Odpowiedź proszę uzasadnić.

26. Napisz makrodefinicję

```
max( A, B, C )
```

zwracającą wartość największą z A,B,C. Np. `max( 1, -1, 0 ) => 1`

27. Dany jest wskaźnik na znak, np.:

```
char *p= "    \t\n\t    ABC    \n\n\n";
```

napisz fragment kodu, który zmodyfikuje wskaźnik p tak, aby wskazywał on na ostatni znak różny od spacji, tabulatora i znaku końca wiersza (kod powinien być niezależny od zawartości ciągu znaków wskazywanego przez p).

28. Napisz **funkcję**

```
char *wstrip( char *s );
```

usuującą wszystkie białe znaki z `s`. Funkcja powinna zwracać wskaźnik do zmodyfikowanego ciągu znaków.

29. Dany jest wskaźnik na znak, np.:

```
char *p= " \t\n\t ABC";
```

napisz fragment kodu obliczający liczbę białych znaków w ciągu wskazywanym przez `p` (kod powinien być niezależny od zawartości ciągu znaków wskazywanego przez `p`). Przyjmij, że ciąg kończy się znakiem `'\0'`.

30. Napisz **funkcję**

```
double min_diff( int n, double v[], double w[] );
```

zwracającą minimalny element różnicy wektorów  $v - w$ , o długości  $n$  każdy.

31. Napisz **funkcję**

```
double l2( int n, double v[] );
```

zwracającą normę  $L_2$  (pierwiastek z sumy kwadratów elementów) wektora  $v$  o długości  $n$ .

32. Aby skompilować pewien program należy wykonać następujące polecenia

```
cc -c project.c
cc -c filesystem.c
cc -o dir project.o filesystem.o
```

Proszę napisać równoważny skrypt dla programu `make`.

33. Pewien program w języku C składa się z trzech plików: `main.c`, `interface.h`, `data.h`. Pliki nagłówkowe `*.h` są wciągane do pliku `main.c`. Poza tym program zależy od pliku `formats.html`. Kompilację programu wykonuje się przy pomocy następującego polecenia:

```
cc -o main.cgi main.c
```

Proszę napisać skrypt dla programu `make`, który będzie tworzył nową wersję `main.cgi` jeżeli od ostatniej kompilacji zmienił się któryś z plików źródłowych.

34. Proszę napisać do czego służą zmienne w skryptach dla programu `make`, i pokazać na przykładzie, jak tych zmiennych używać.
35. Proszę opisać, jak wygląda reguła w skryptach dla programu `make`, i pokazać na przykładzie, jej elementy składowe.