

# Function (1A)

---

Copyright (c) 2010, 2011 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to [youngwlim@hotmail.com](mailto:youngwlim@hotmail.com).

This document was produced by using OpenOffice.

# Task: Finding Partial Sums (1)

$$S_n = \sum_{k=1}^n a_k$$

$$a_k = k$$

$$S_1 = \sum_{k=1}^1 k = 1$$

```
printf("S1 = %d \n", S1);
```

$$S_2 = \sum_{k=1}^2 k = 1 + 2$$

```
printf("S2 = %d \n", S2);
```

$$S_3 = \sum_{k=1}^3 k = 1 + 2 + 3$$

```
printf("S3 = %d \n", S3);
```

# Task: Finding Partial Sums (2)

$$S_1 = \sum_{k=1}^1 k = 1$$

```
S1 = 0;  
for (k=1; k<=1; ++k) S1 += k;
```

```
printf("S1 = %d \n", S1);
```

$$S_2 = \sum_{k=1}^2 k = 1 + 2$$

```
S2 = 0;  
for (k=1; k<=2; ++k) S2 += k;
```

```
printf("S2 = %d \n", S2);
```

$$S_3 = \sum_{k=1}^3 k = 1 + 2 + 3$$

```
S3 = 0;  
for (k=1; k<=3; ++k) S3 += k;
```

```
printf("S3 = %d \n", S3);
```

# Task: Finding Partial Sums (3)

```
input_n = 1;
```

```
{  
    int n = input_n;  
    int S = 0;  
    for (k=1; k<=n; ++k) S += k;  
}
```

```
S1 = S;
```

```
printf("S1 = %d \n", S1);
```

```
input_n = 2;
```

```
{  
    int n = input_n;  
    int S = 0;  
    for (k=1; k<=n; ++k) S += k;  
}
```

```
S2 = S;
```

```
printf("S2 = %d \n", S2);
```

```
input_n = 3;
```

```
{  
    int n = input_n;  
    int S = 0;  
    for (k=1; k<=n; ++k) S += k;  
}
```

```
S3 = S;
```

```
printf("S3 = %d \n", S3);
```

```
int psum (int n)
```

```
{  
    int S = 0;  
    for (k=1; k<=n; ++k) S += k;  
    return S;  
}
```

# Task: Finding Partial Sums (4)

```
S1 = psum ( 1 );
```

```
printf("S1 = %d \n", S1);
```

```
S2 = psum ( 2 );
```

```
printf("S1 = %d \n", S2);
```

```
S3 = psum ( 3 );
```

```
printf("S1 = %d \n", S3);
```

```
int psum (int n)
```

```
{  
    int S = 0;  
    for (k=1; k<=n; ++k) S += k;  
    return S;  
}
```

input\_n = 1;

input\_n = 2;

input\_n = 3;

S1 = S;

S1 = S<sub>i</sub>;

S2 = S<sub>i</sub>;

S1 = S<sub>i</sub>;

```
input_n = ;
```

```
{  
    int n = input_n;  
    int S = 0;  
    for (k=1; k<=n; ++k) S += k;  
}
```

● = S;

# Function Prototype (1)

```
int psum (int n) ;
```

*To inform the compiler that **psum** is the name of a function which has one integer type input and whose output type is integer*

```
int main (void)
```

```
{  
    int S1, S2, S3;  
  
    S1 = psum ( 1 );  
    printf("S1 = %d \n", S1);  
    S2 = psum ( 2 );  
    printf("S2 = %d \n", S2);  
    S3 = psum ( 3 );  
    printf("S3 = %d \n", S3);  
  
    return 0;  
}
```

*Since **psum** identifier is declared, **psum** can be used here.*

```
int psum (int n)
```

```
{  
    int S = 0;  
    for (k=1; k<=n; ++k) S += k;  
    return S;  
}
```

*What the function **psum** actually does is defined here.*

# Function Prototype (2)

```
int psum (int n)
{
    int S = 0;
    for (k=1; k<=n; ++k) S += k;
    return S;
}
```

→ The function *psum* is defined here.

```
int main (void)
{
    int S1, S2, S3;

    S1 = psum ( 1 );
    printf("S1 = %d \n", S1);
    S2 = psum ( 2 );
    printf("S2 = %d \n", S2);
    S3 = psum ( 3 );
    printf("S3 = %d \n", S3);

    return 0;
}
```

← Since *psum* identifier is declared, *psum* can be used here.



# Function Prototype (3)

src1.c

```
int psum (int n) ;
```

```
int main (void)
```

```
{  
    int S1, S2, S3;  
  
    S1 = psum ( 1 );  
    printf("S1 = %d \n", S1);  
    S2 = psum ( 2 );  
    printf("S2 = %d \n", S2);  
    S3 = psum ( 3 );  
    printf("S3 = %d \n", S3);  
  
    return 0;  
}
```

```
int psum (int n)
```

```
{  
    int S = 0;  
    for (k=1; k<=n; ++k) S += k;  
    return S;  
}
```

src2.c

```
int psum (int n)
```

```
{  
    int S = 0;  
    for (k=1; k<=n; ++k) S += k;  
    return S;  
}
```

```
int main (void)
```

```
{  
    int S1, S2, S3;  
  
    S1 = psum ( 1 );  
    printf("S1 = %d \n", S1);  
    S2 = psum ( 2 );  
    printf("S2 = %d \n", S2);  
    S3 = psum ( 3 );  
    printf("S3 = %d \n", S3);  
  
    return 0;  
}
```

# Function Prototype (4)

src3.c

```
int psum (int n) ;
```

```
int main (void)
```

```
{  
    int S1, S2, S3;  
  
    S1 = psum ( 1 );  
    printf("S1 = %d \n", S1);  
    S2 = psum ( 2 );  
    printf("S2 = %d \n", S2);  
    S3 = psum ( 3 );  
    printf("S3 = %d \n", S3);  
  
    return 0;  
}
```

src4.c

```
int psum (int n)
```

```
{  
    int S = 0;  
    for (k=1; k<=n; ++k) S += k;  
    return S;  
}
```

# Function Prototype (5)

src5.c

```
#include "src4.h"

int main (void)
{
    int S1, S2, S3;

    S1 = psum ( 1 );
    printf("S1 = %d \n", S1);
    S2 = psum ( 2 );
    printf("S2 = %d \n", S2);
    S3 = psum ( 3 );
    printf("S3 = %d \n", S3);

    return 0;
}
```

src4.c

```
int psum (int n)
{
    int S = 0;
    for (k=1; k<=n; ++k) S += k;
    return S;
}
```

src4.h

```
int psum (int n) ;
```

## References

- [1] Essential C, input\_nck Parlante
- [2] Efficient C Programming, Mark A. Weiss
- [3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
- [4] C Language Express, I. K. Chun